CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Peer-to-Peer Networks

13 Internet – The Underlay Network

Christian Ortolf
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

UNI FREIBURG

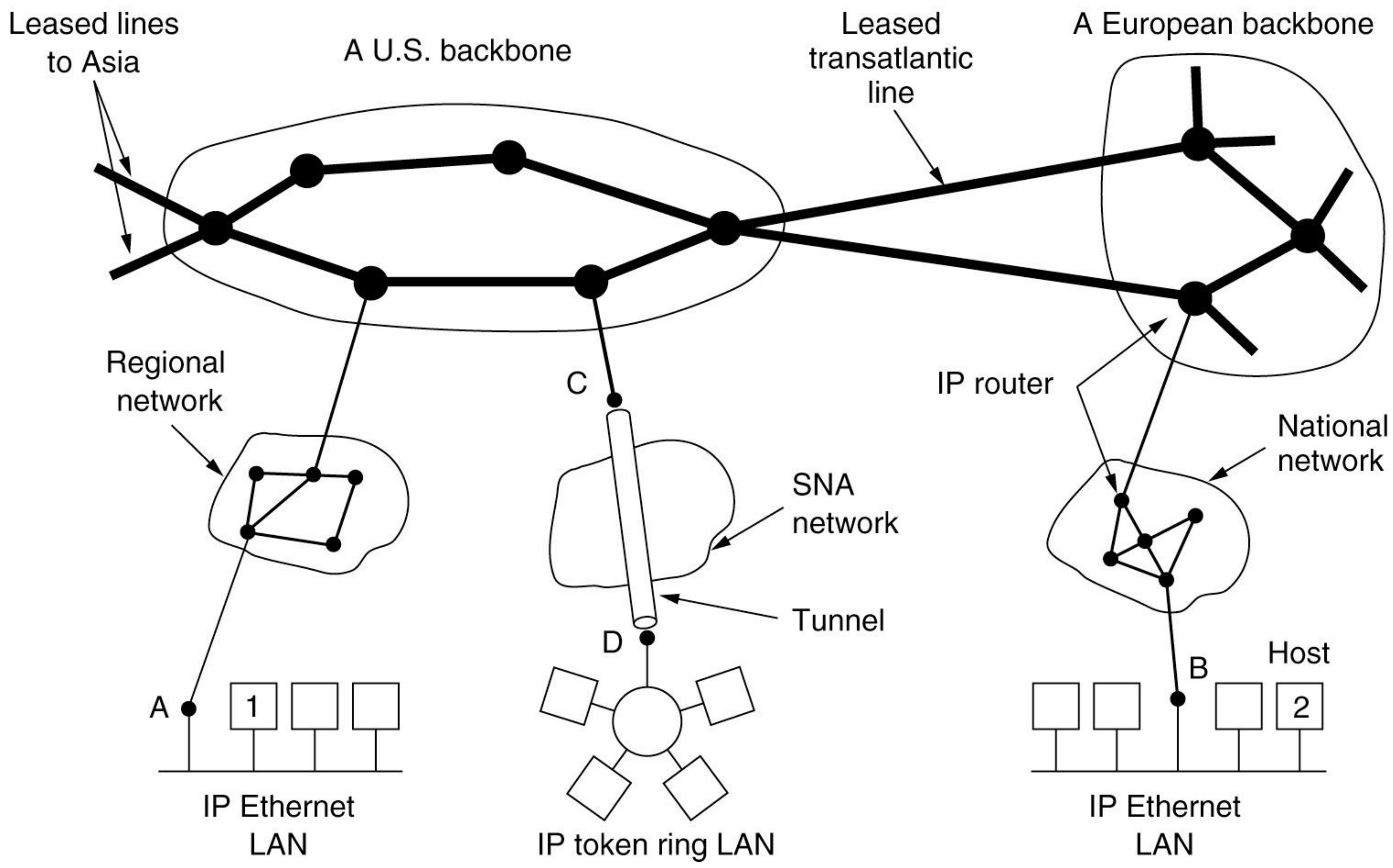
Types of Networks

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	Local area network
1 km	Campus	Local area network
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	Wide area network
10,000 km	Planet	The Internet

(Tanenbaum)

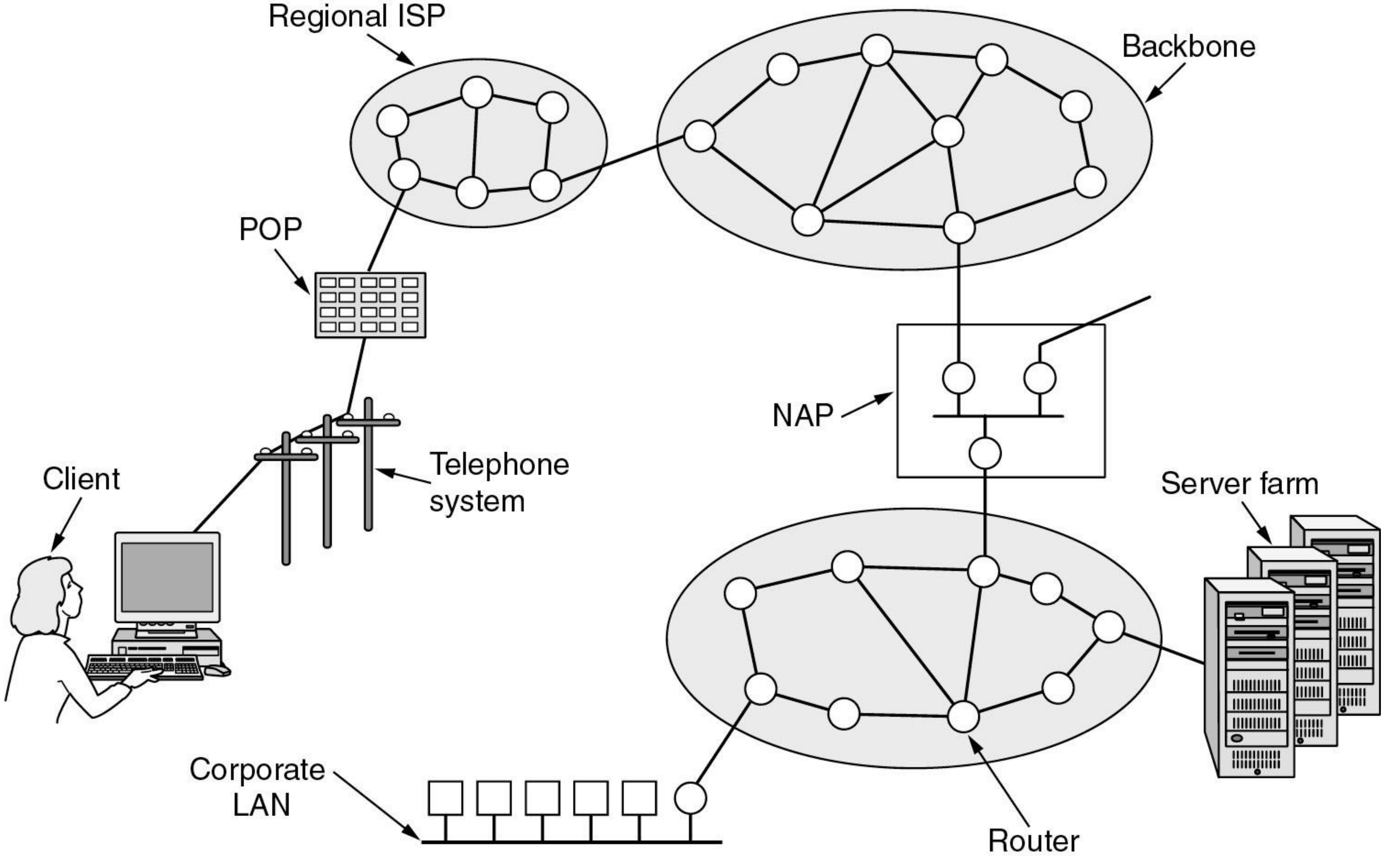
The Internet

- global system of interconnected WANs and LANs
- open, system-independent, no global control



[Tanenbaum, Computer Networks]

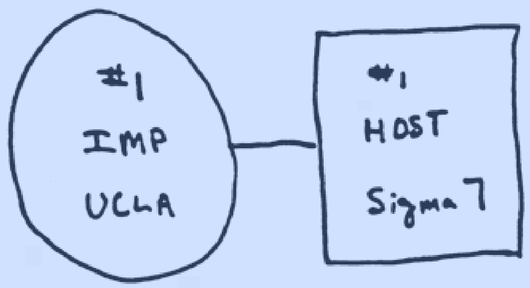
Interconnection of Subnetworks



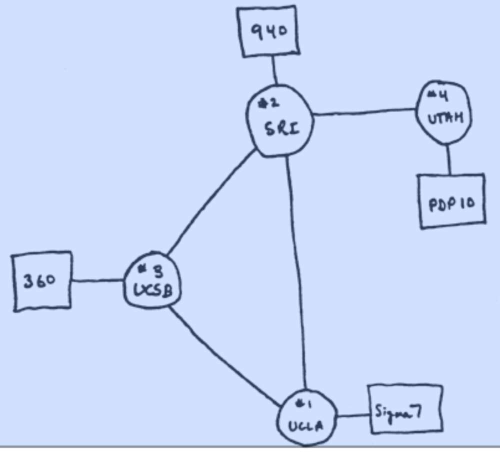
[Tanenbaum, Computer Networks]

History of the Internet

- 1961: Packet Switching Theory
  - Leonard Kleinrock, MIT, “Information Flow in Communication Nets”
- 1962: Concept of a “Galactic Network”
  - J.C.R. Licklider and W. Clark, MIT, “On-Line Man Computer Communication”
- 1965: Predecessor of the Internet
  - Analog modem connection between 2 computers in the USA
- 1967: Concept of the “ARPANET”
  - Concept of Larry Roberts
- 1969: 1st node of the “ARPANET”
  - at UCLA (Los Angeles)
  - end 1969: 4 computers connected

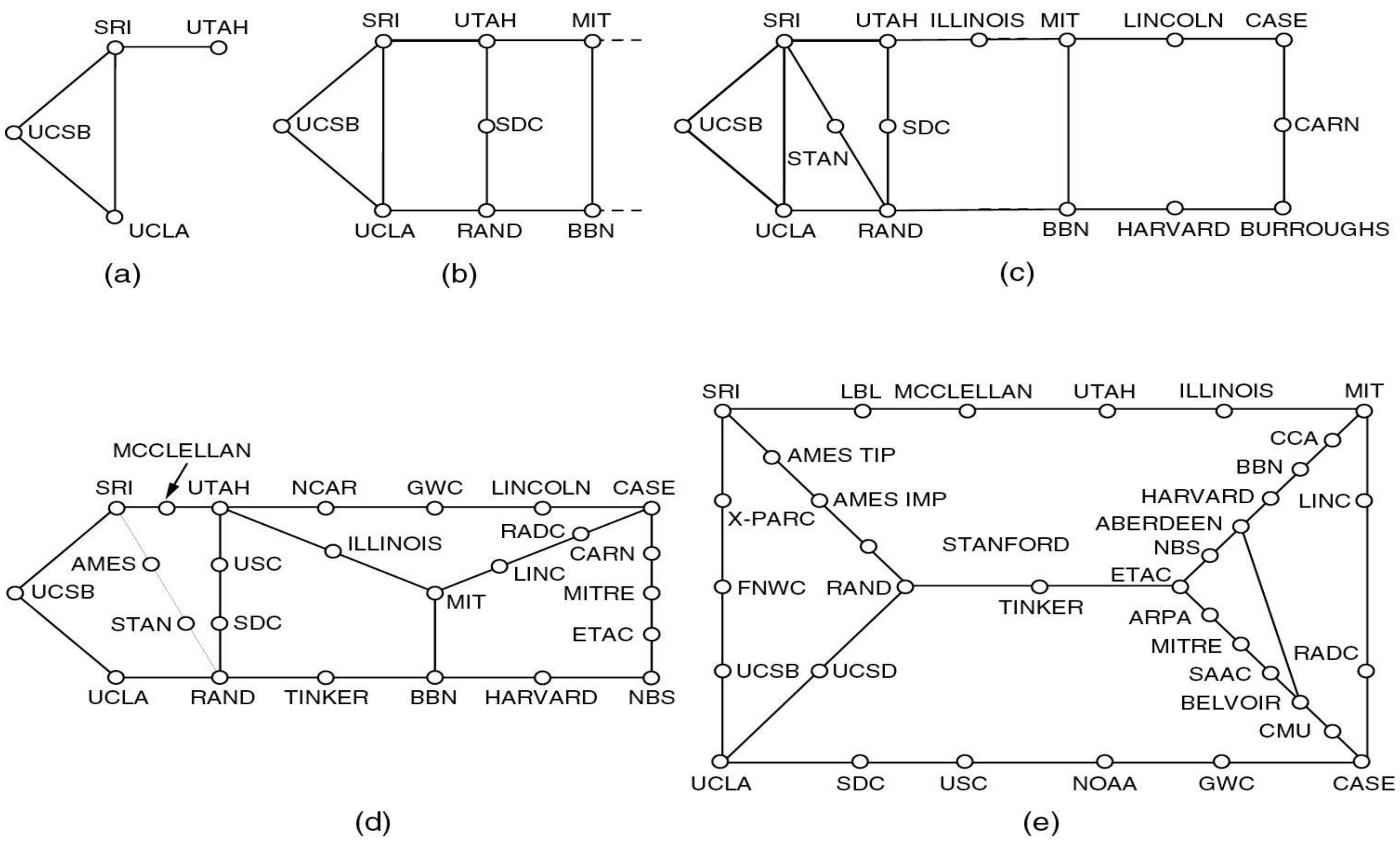


conceptual sketches of the original internet

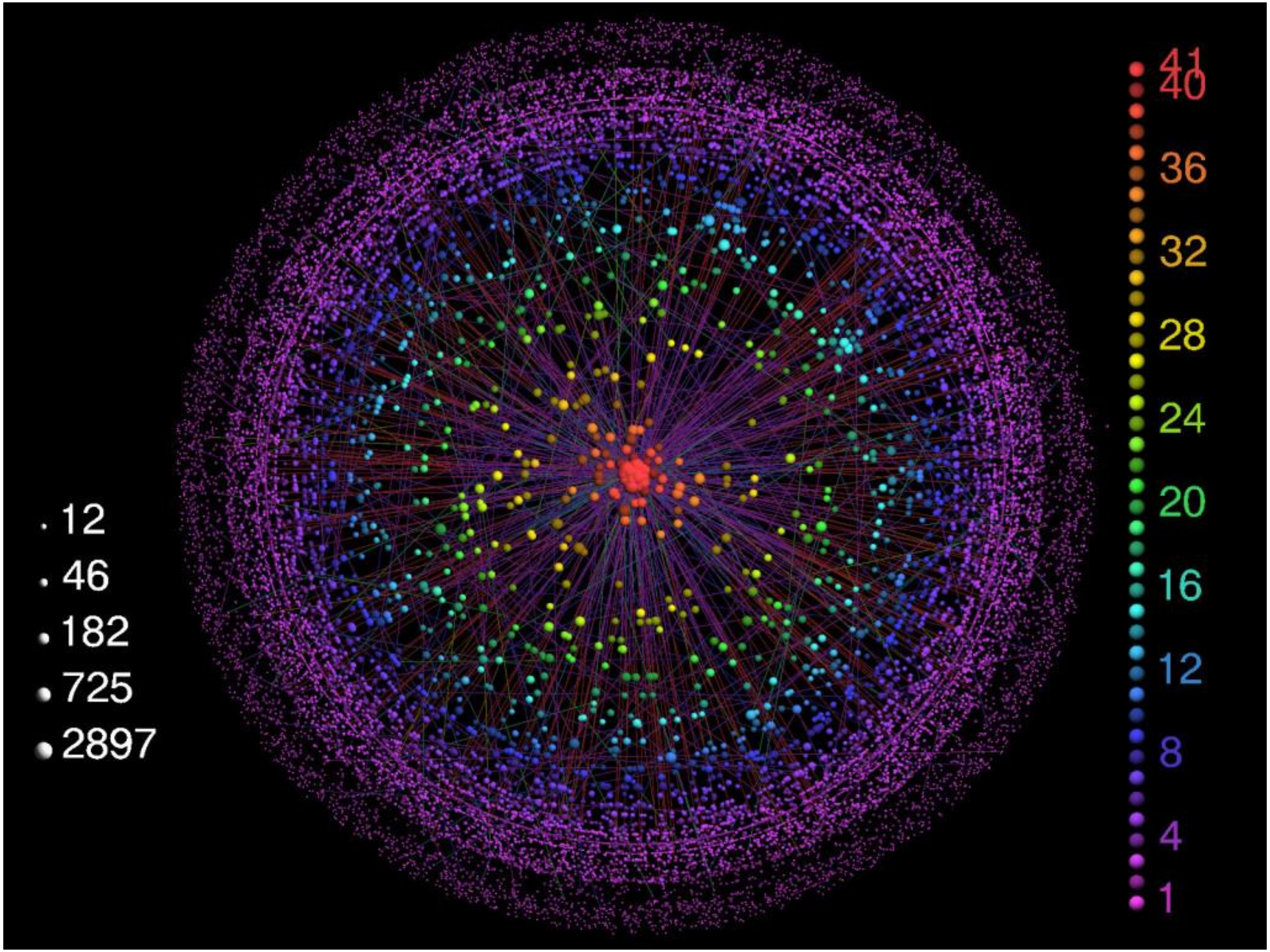


ARPANET

ARPANET (a) December 1969 (b) July 1970
(c) March 1971 (d) April 1972 (e) September 1972



Internet ~2005



An Open Network Architecture

- Concept of Robert Kahn (DARPA 1972)
  - Local networks are autonomous
    - independent
    - no WAN configuration
  - packet-based communication
  - “best effort” communication
    - if a packet cannot reach the destination, it will be deleted
    - the application will re-transmit
  - black-box approach to connections
    - black boxes: gateways and routers
    - packet information is not stored
    - no flow control
  - no global control
- Basic principles of the Internet

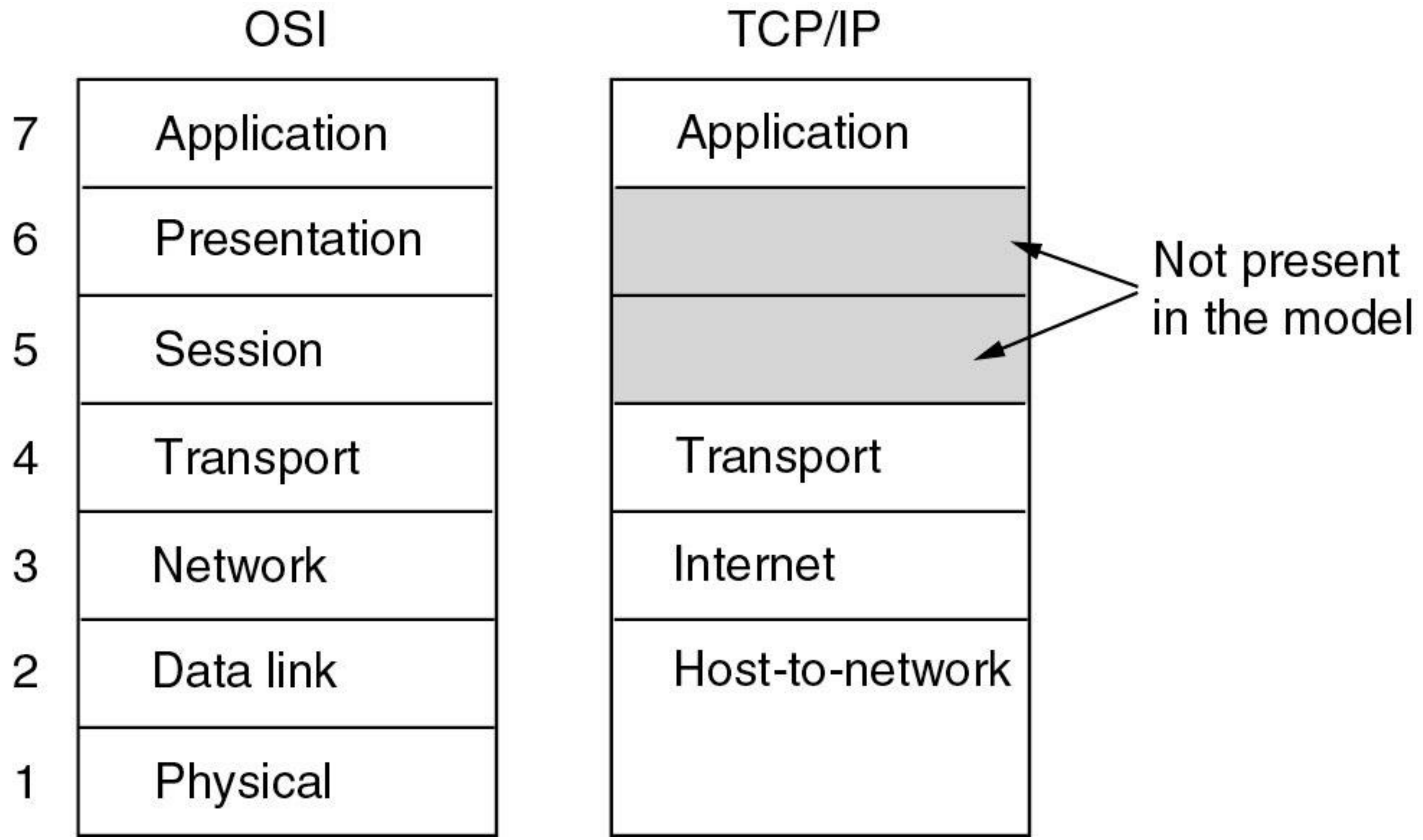
Protocols of the Internet

Application	Telnet, FTP, HTTP, SMTP (E-Mail), ...
Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Network	IP (Internet Protocol) IPv4 + IPv6 + ICMP (Internet Control Message Protocol) + IGMP (Internet Group Management Protoccol)
Host-to-Network	LAN (e.g. Ethernet, W-Lan etc.)

TCP/IP Layers

- 1. Host-to-Network
  - Not specified, depends on the local networ,k e.g. Ethernet, WLAN 802.11, PPP, DSL
- 2. Routing Layer/Network Layer (IP - Internet Protocol)
  - Defined packet format and protocol
  - Routing
  - Forwarding
- 3. Transport Layer
  - TCP (Transmission Control Protocol)
    - Reliable, connection-oriented transmission
    - Fragmentation, Flow Control, Multiplexing
  - UDP (User Datagram Protocol)
    - hands packets over to IP
    - unreliable, no flow control
- 4. Application Layer
  - Services such as TELNET, FTP, SMTP, HTTP, NNTP (for DNS), …
  - Peer-to-peer networks

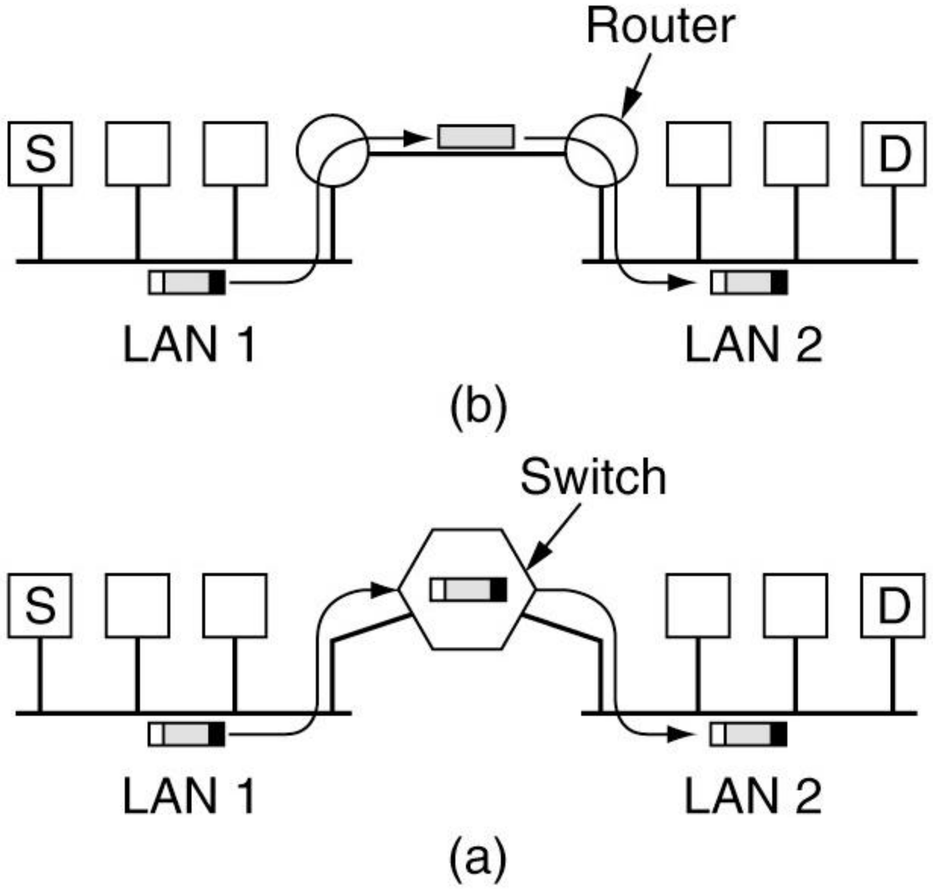
Reference Models: OSI versus TCP/IP



(Aus Tanenbaum)

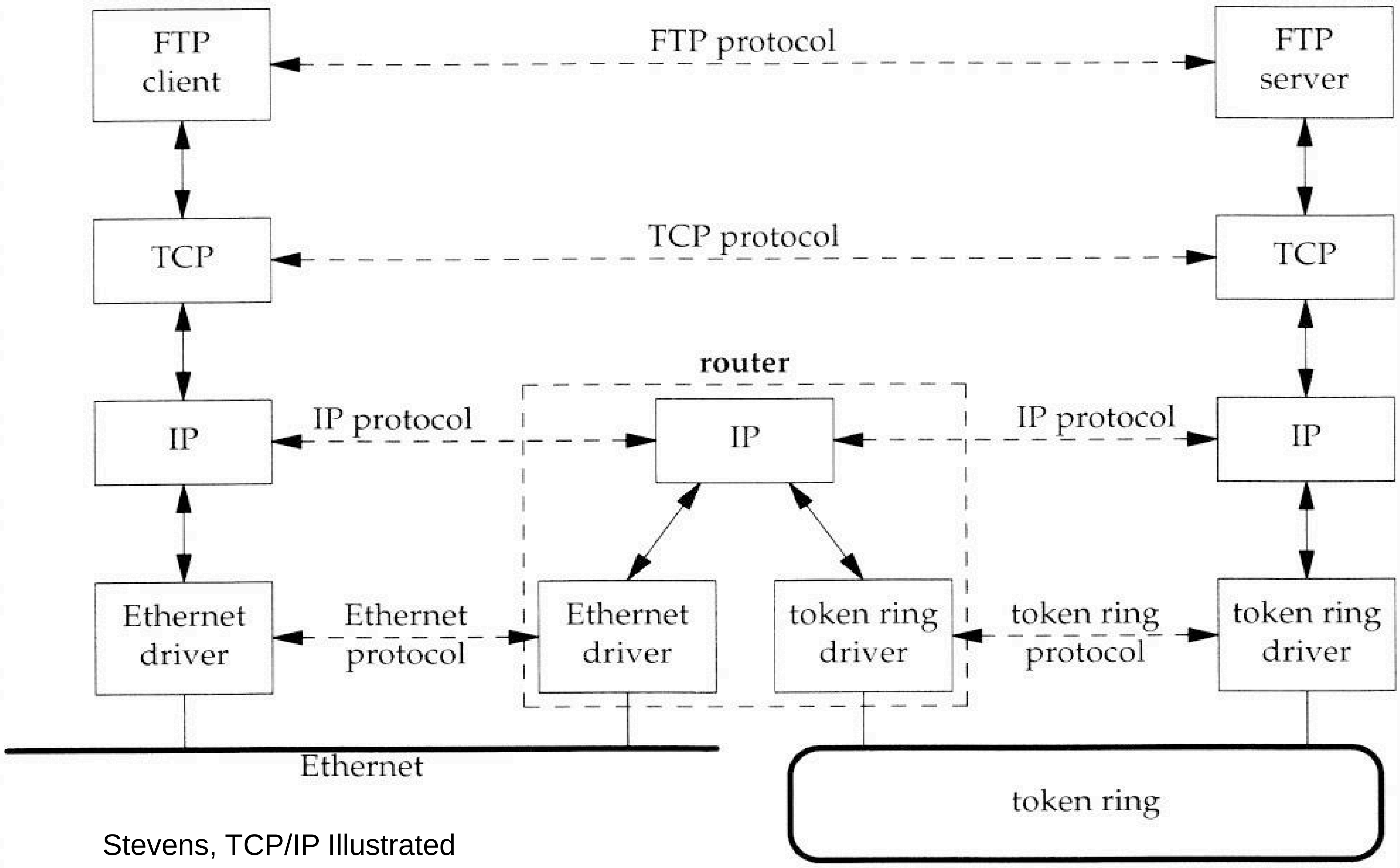
Network Interconnections

Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub



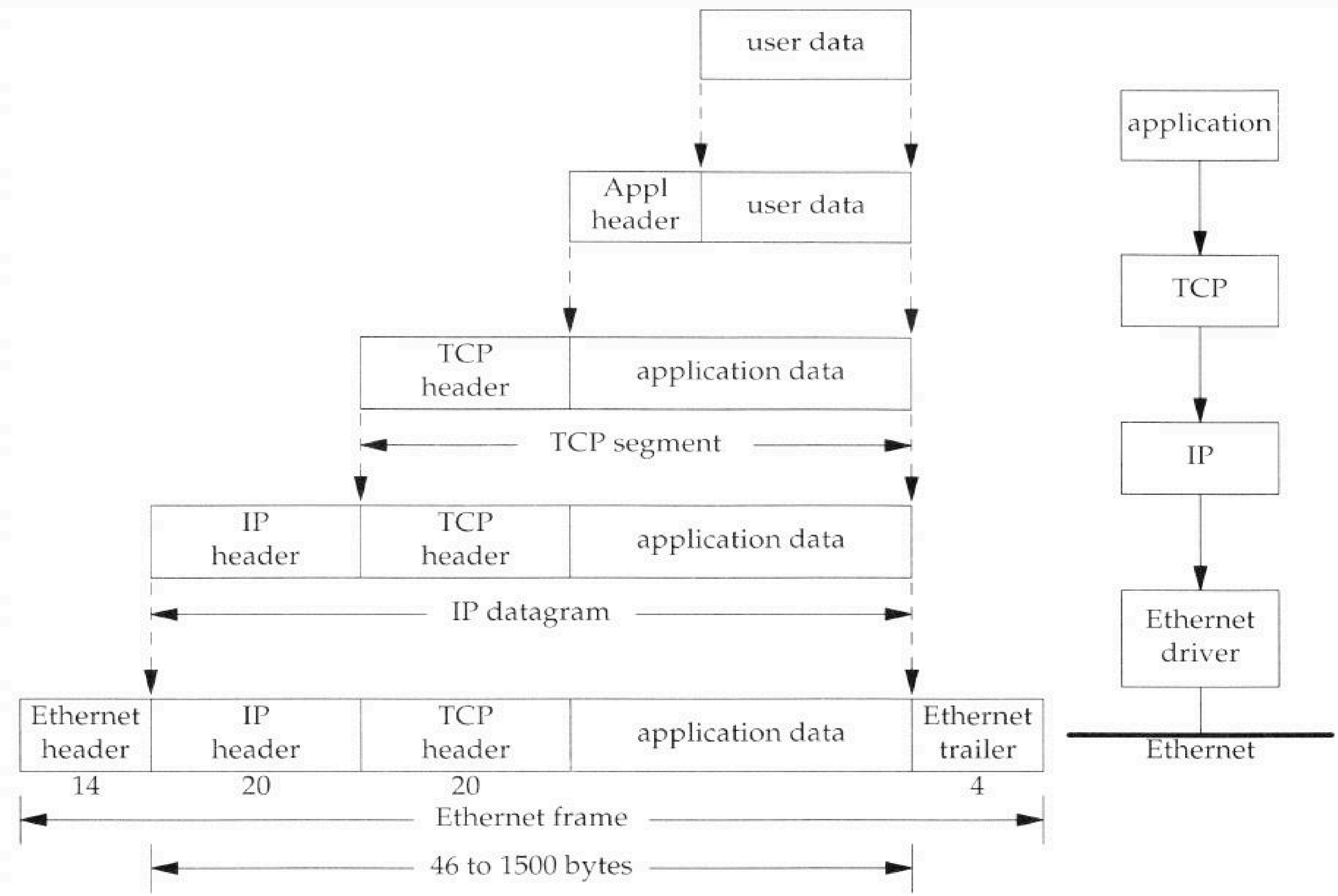
[Tanenbaum, Computer Networks]

Example: Routing between LANs



Stevens, TCP/IP Illustrated

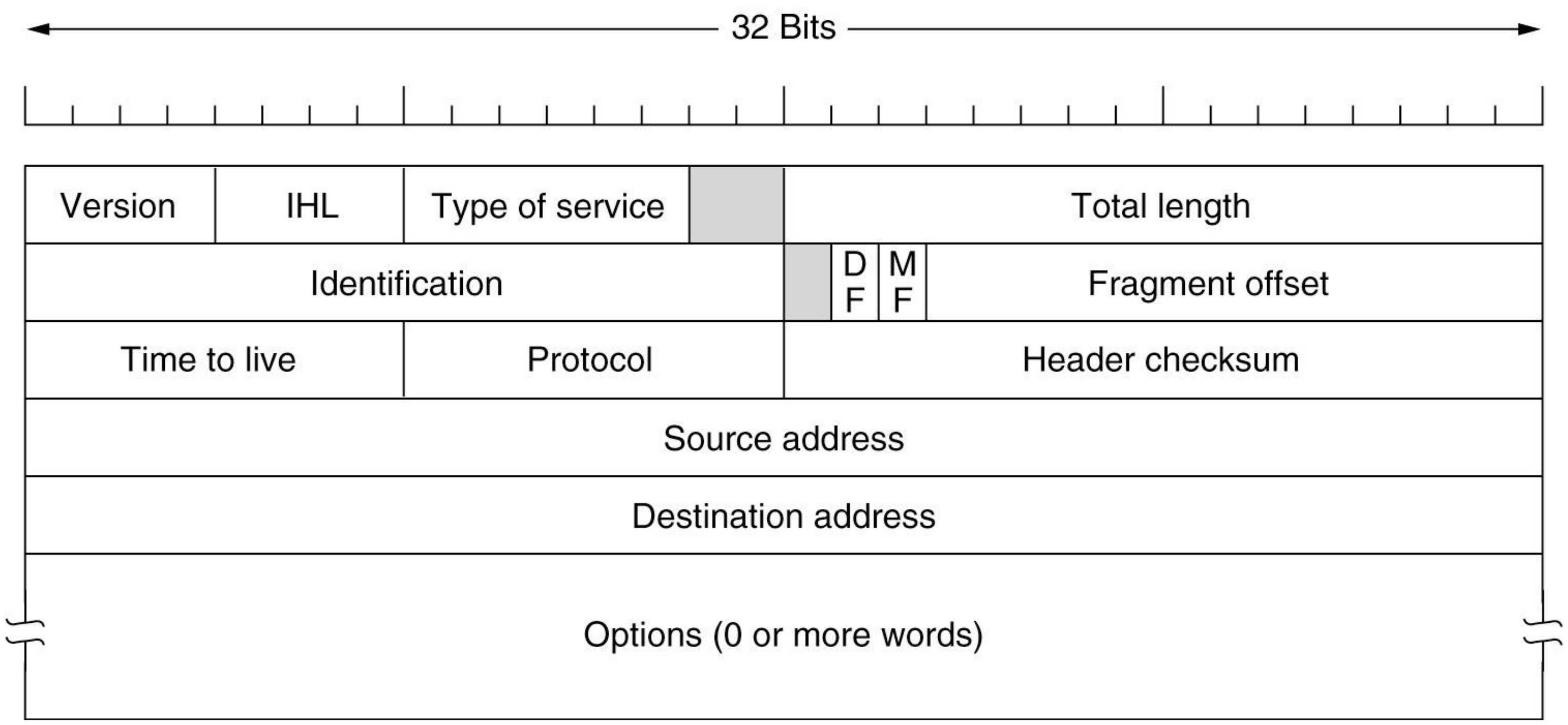
Data/Packet Encapsulation



Stevens, TCP/IP Illustrated

IPv4-Header (RFC 791)

- Version: 4 = IPv4
- IHL: IP header length
  - in 32 bit words (>5)
- Type of service
  - optimize delay, throughput, reliability, monetary cost
- Checksum (only IP-header)
- Source and destination IP-address
- Protocol identifies protocol
  - e.g. TCP, UDP, ICMP, IGMP
- Time to Live:
  - maximal number of hops

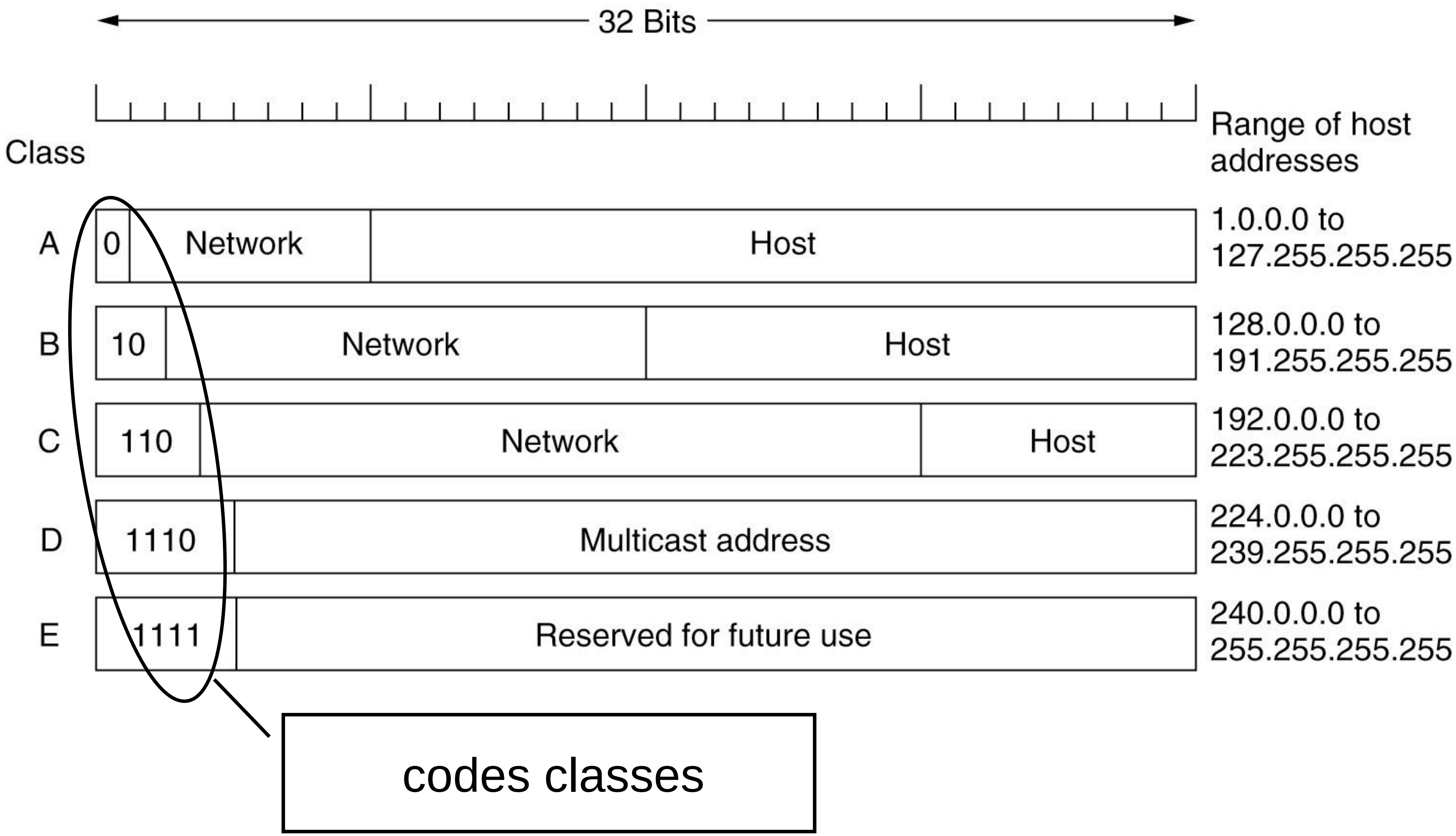


IP addresses and Domain Name System

- IP addresses
  - every interface in a network has a unique world wide IP address
  - separated in Net-ID and Host-ID
  - Net-ID assigned by Internet Network Information Center
  - Host-ID by local network administration
- Domain Name System (DNS)
  - replaces IP addresses like 132.230.167.230 by names, e.g. falcon.informatik.uni-freiburg.de and vice versa
  - Robust distributed database

Internet IP Addresses
Classfull Addresses until 1993

- Classes A, B, and C
- D for multicast; E: “reserved”



128 NWs; 16 M hosts

16K NWs; 64K hosts

2M NWs; 256 hosts

Classless IPv4-Addresses

- Until 1993 (deprecated)
  - 5 classes marked by Präfix
  - Then sub-net-id prefix of fixed length and host-id
- Since 1993
  - Classless Inter-Domain-Routing (CIDR)
  - Net-ID and Host-ID are distributed flexibly
  - E.g.
    - Network mask /24 or 11111111.11111111.11111111.00000000
    - denotes, that IP-address
      - 10000100. 11100110. 10010110. 11110011
      - consists of network 10000100. 11100110. 10010110
      - and host 11110011
- Route aggregation
  - Routing protocols BGP, RIP v2 and OSPF can address multiple networks using one ID
    - Z.B. all Networks with ID 10010101010* can be reached over host X

Routing Tables and Packet Forwarding

- IP Routing Table
  - contains for each destination the address of the next gateway
  - destination: host computer or sub-network
  - default gateway
- Packet Forwarding
  - IP packet (datagram) contains start IP address and destination IP address
    - if destination = my address then hand over to higher layer
    - if destination in routing table then forward packet to corresponding gateway
    - if destination IP subnet in routing table then forward packet to corresponding gateway
    - otherwise, use the default gateway

IP Packet Forwarding

- IP -Packet (datagram) contains...
  - TTL (Time-to-Live): Hop count limit
  - Start IP Address
  - Destination IP Address
- Packet Handling
  - Reduce TTL (Time to Live) by 1
  - If TTL ≠ 0 then forward packet according to routing table
  - If TTL = 0 or forwarding error (buffer full etc.):
    - delete packet
    - if packet is not an ICMP Packet then
      - send ICMP Packet with
      - start = current IP Address
      - destination = original start IP Address

Introduction to Future IP

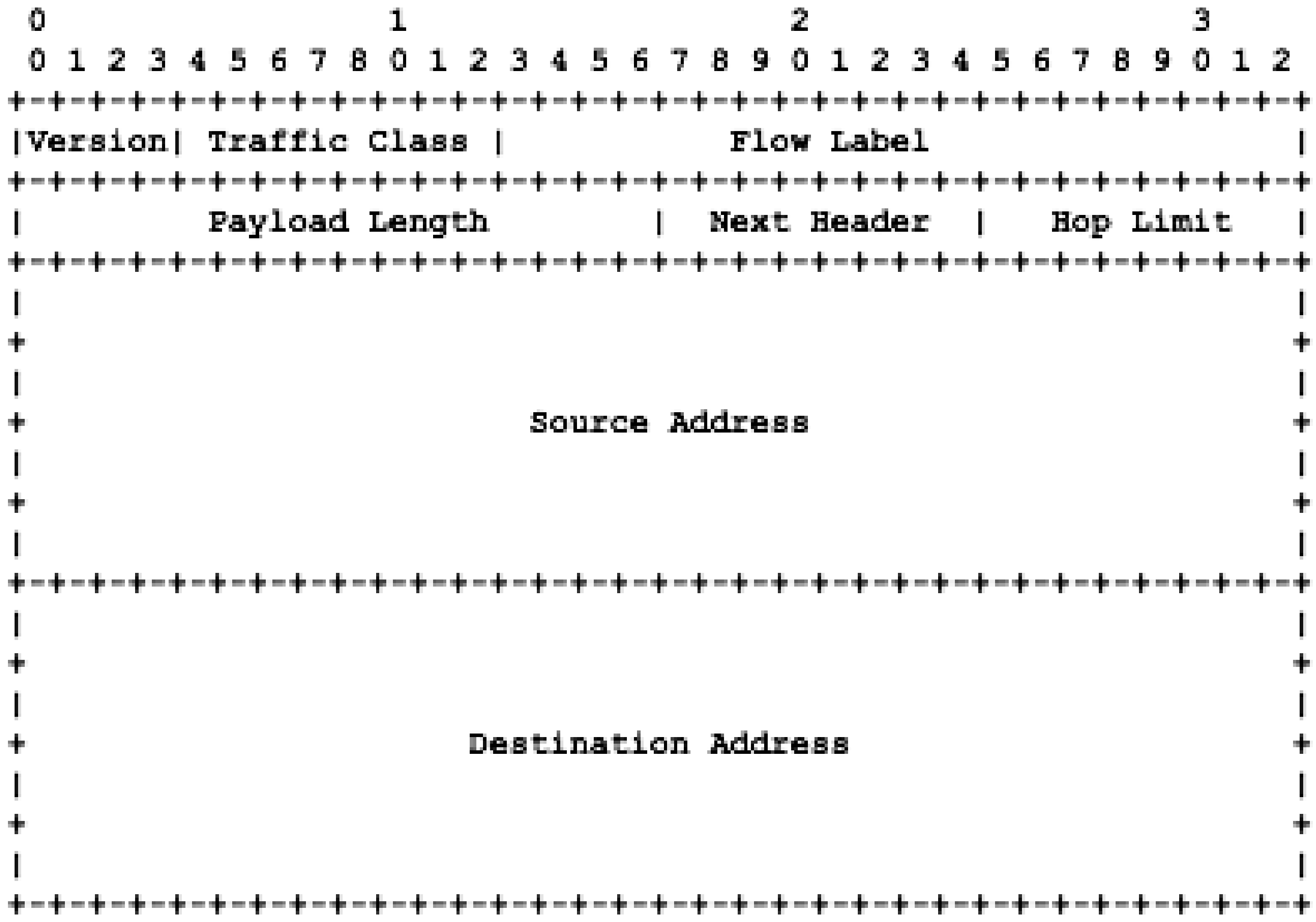
- IP version 6 (IP v6 – around July 1994)
- Why switch?
  - rapid, exponential growth of networked computers
  - shortage (limit) of the addresses
  - new requirements towards the Internet infrastructure (streaming, real-time services like VoIP, video on demand)
- evolutionary step from IPv4
- interoperable with IPv4

Capabilities of IP

- dramatic changes of IP
  - Basic principles still appropriate today
  - Many new types of hardware
  - Scale of Internet and interconnected computers in private LAN
- Scaling
  - Size - from a few tens to a few tens of millions of computers
  - Speed - from 9,6Kbps (GSM) to 10Gbps (Ethernet)
  - Increased frame size (MTU) in hardware

IPv6-Header (RFC 2460)

- Version: 6 = IPv6
- Traffic Class
  - for QoS (priority)
- Flow Label
  - QoS or real-time
- Payload Length
  - size of the rest of the IP packet
- Next Header (IPv4: protocol)
  - e..g. ICMP, IGMP, TCP, EGP, UDP, Multiplexing, ...
- Hop Limit (Time to Live)
  - maximum number of hops
- Source Address
- Destination Address
  - 128 bit IPv6 address



Static and Dynamic Routing

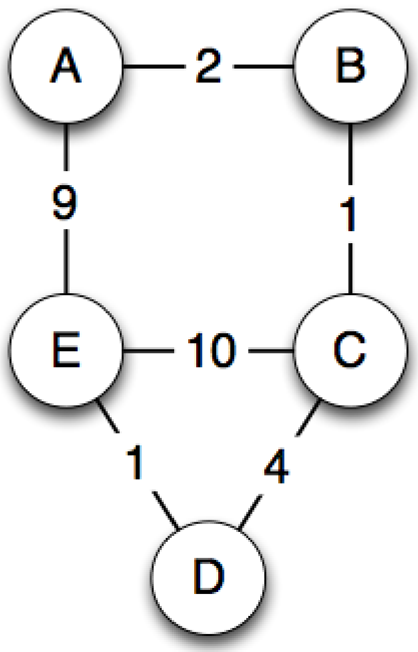
- Static Routing
  - Routing table created manually
  - used in small LANs
- Dynamic Routing
  - Routing table created by Routing Algorithm
  - Centralized, e.g. Link State
    - Router knows the complete network topology
  - Decentralized, e.g. Distance Vector
    - Router knows gateways in its local neighborhood

Intra-AS Routing

- Routing Information Protocol (RIP)
  - Distance Vector Algorithmus
  - Metric = hop count
  - exchange of distance vectors (by UDP)
- Interior Gateway Routing Protocol (IGRP)
  - successor of RIP
  - different routing metrics (delay, bandwidth)
- Open Shortest Path First (OSPF)
  - Link State Routing (every router knows the topology)
  - Route calculation by Dijkstra’s shortest path algorithm

Distance Vector Routing Protocol

- Distance Table data structure
  - Each node has a
    - Line for each possible destination
    - Column for any direct neighbors
- Distributed algorithm
  - each node communicates only with its neighbors
- Asynchronous operation
  - Nodes do not need to exchange information in each round
- Self-terminating
  - exchange unless no update is available



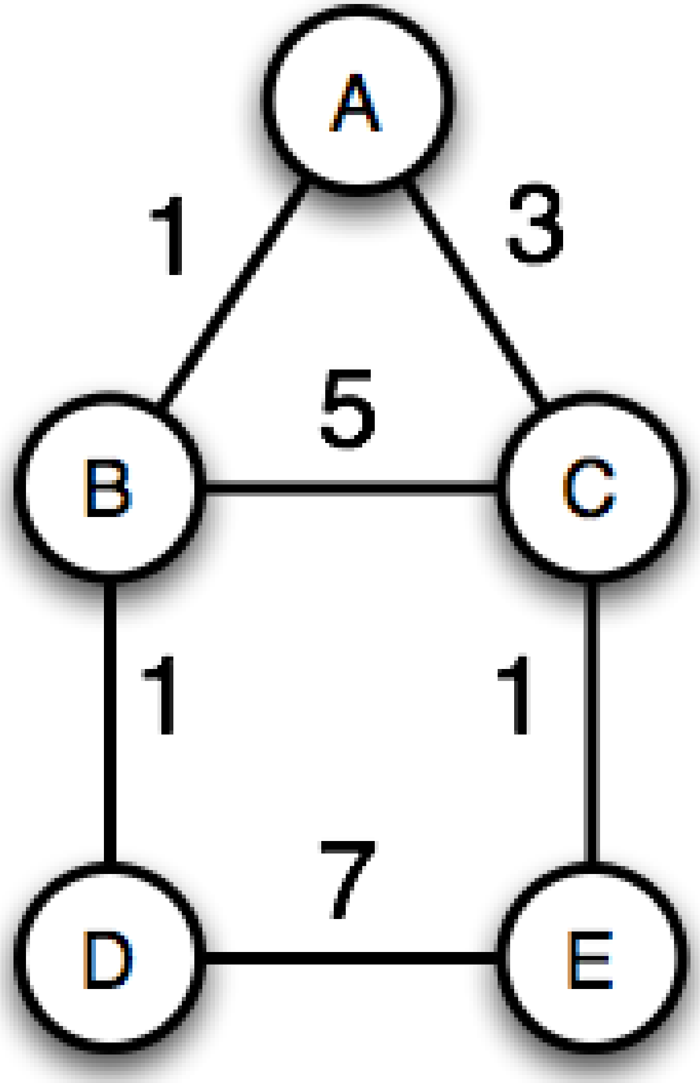
Distance Table for A

from A	via B	via E	Routing Table entry
to B	**2**	15	B
C	**3**	14	B
D	**7**	10	B
E	**8**	9	E

Distance Table for C

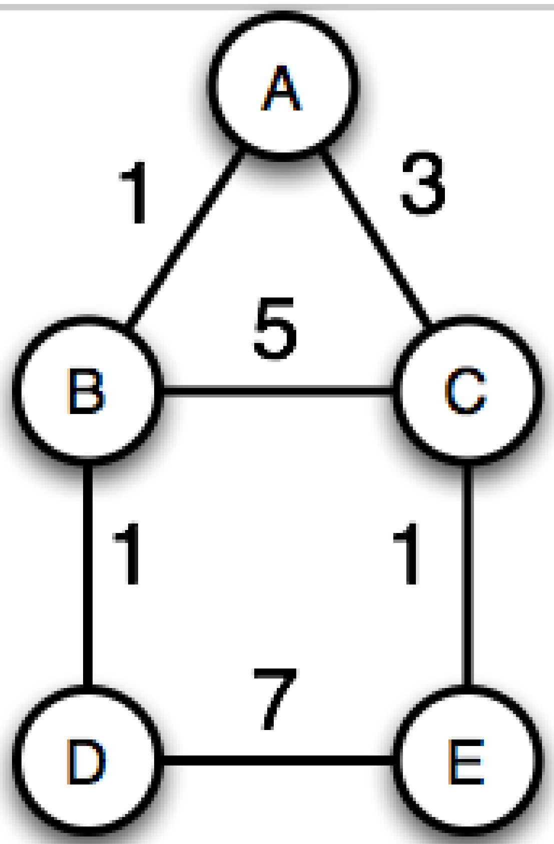
from C	via B	via D	via E	Routing Table entry
to A	**3**	11	18	B
B	**1**	9	21	B
D	6	**4**	11	D
E	7	**5**	10	D

Distance Vector Routing Example



from A to	via B	via C	entry
B	**1**	8	**B**
C	6	**3**	**C**
D	**2**	9	**B**
E	7	**4**	**C**

Distance Vector Routing



from A to	via B	via C	entry
B	1	-	B
C	-	3	C
D	-	-	-
E	-	-	-

from B to	via A	via C	via D	entry
A	1	-	-	A
C	-	3	-	C
D	-	-	1	C
E	-	-	8	D

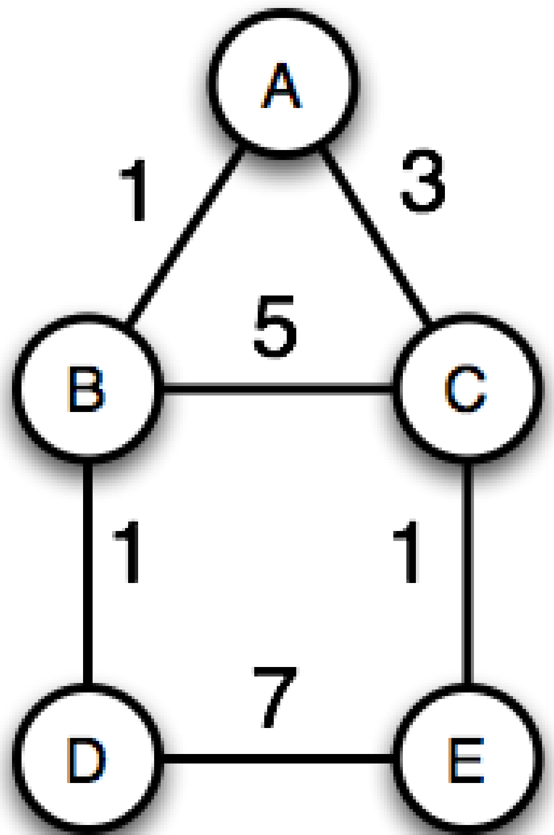
from C to	via A	via B	via E	entry
A	3	-	-	A
B	-	5	-	B
D	-	-	8	E
E	-	-	1	E

Distance Vector Routing

from B to	via A	via C	via D	Entry
A	1	-	-	A
C	-	5	-	C
D	-	-	1	D
E	-	-	8	D

from C to	via A	via B	via E	Entry
A	3	-	-	A
B	-	5	-	B
D	-	-	8	E
E	-	-	1	E

from B to	via A	via C	via D	Entry
A	1	8	-	A
C	-	5	-	C
D	-	13	1	D
E	-	6	8	C

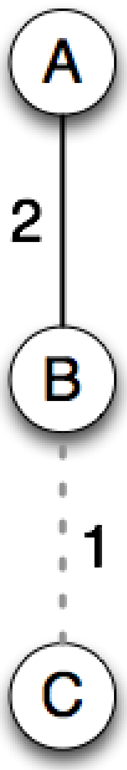


from C to	via A	via B	via E	Entry
A	3	6	-	A
B	-	5	-	B
D	-	6	8	B
E	-	13	1	E

“Count to Infinity” - Problem

- Good news travels fast
  - A new connection is quickly at hand
- Bad news travels slowly
  - Connection fails
  - Neighbors increase their distance mutally
  - "Count to Infinity" Problem

"Count to Infinity" - Problem



from A		via B	Routing Table entry
to	B	2	B
	C	3	B

from B		via A	via C	Routing Table entry
to	A	2	-	A
	C	5	-	A

from A		via B	Routing Table entry
to	B	2	B
	C	7	B

from B		via A	via C	Routing Table entry
to	A	2	-	A
	C	5	-	A

from A		via B	Routing Table entry
to	B	2	B
	C	7	B

from B		via A	via C	Routing Table entry
to	A	2	-	A
	C	9	-	A

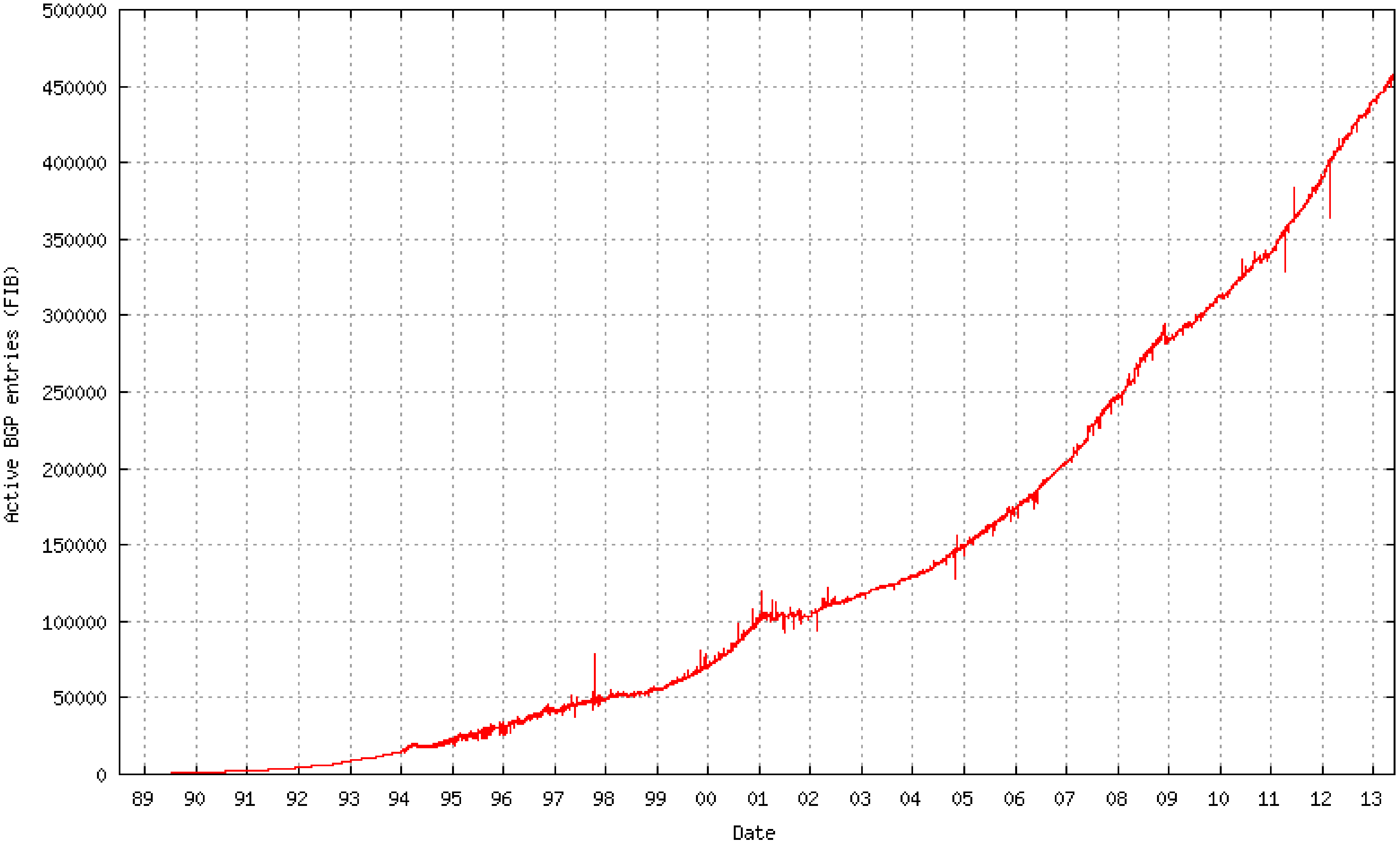
Link-State Protocol

- Link state routers
  - exchange information using Link State Packets (LSP)
  - each node uses shortest path algorithm to compute the routing table
- LSP contains
  - ID of the node generating the packet
  - Cost of this node to any direct neighbors
  - Sequence-no. (SEQNO)
  - TTL field for that field (time to live)
- Reliable flooding (Reliable Flooding)
  - current LSP of each node are stored
  - Forward of LSP to all neighbors
    - except to be node where it has been received from
  - Periodically creation of new LSPs
    - with increasing SEQNO
  - Decrement TTL when LSPs are forwarded

Inter-AS: BGPv4 (Border Gateway Protocol)

- de facto standard
- Path-Vector-Protocol
  - like Distance Vector Protocol
    - store whole path to the target
  - each Border Gateway advertizes to all its neighbors (peers) the complete path to the target (per TCP)
- If gateway X sends the path to the peer-gateway W
  - then W can choose the path or not
  - optimization criteria
    - cost, policy, etc.
  - if W chooses the path of X, it publishes
    - Path(W,Z) = (W, Path (X,Z))
- Remark
  - X can control incoming traffic using advertisements
  - all details hidden here

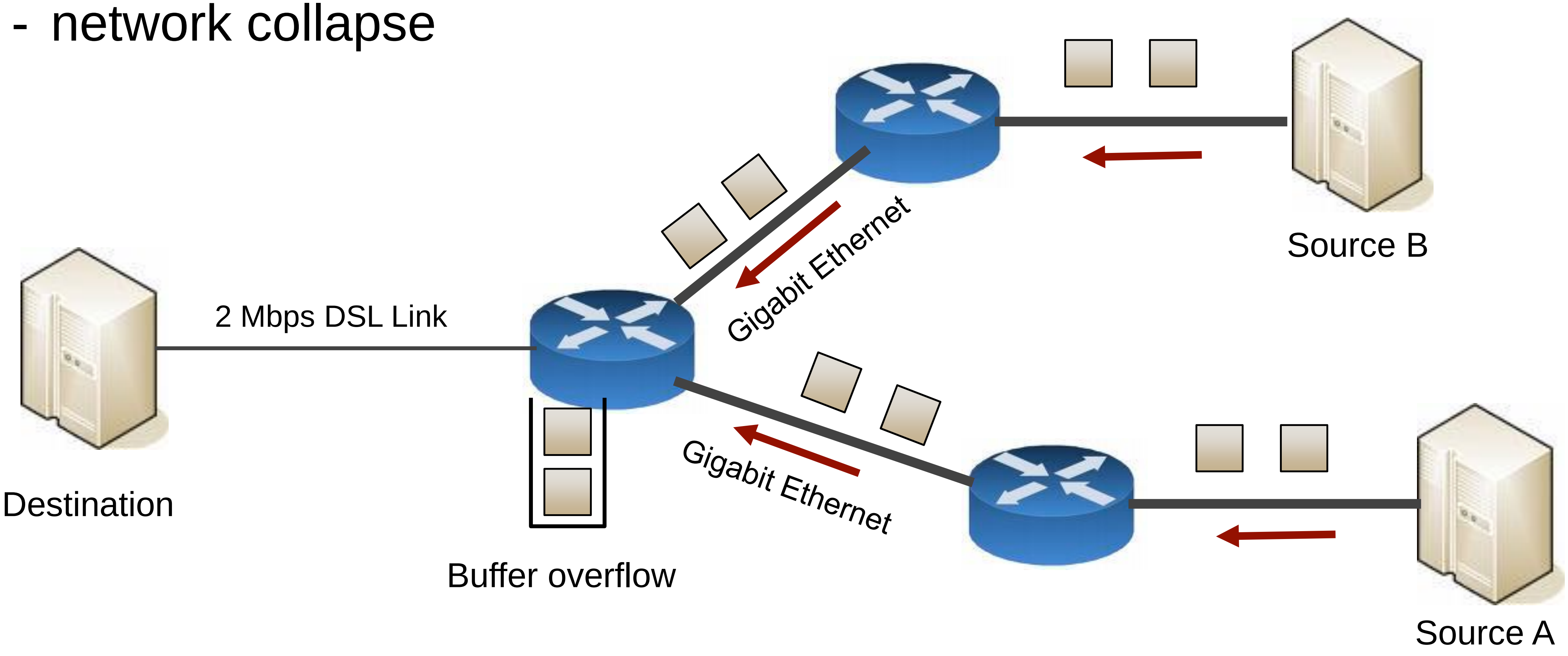
BGP-Routing Table Size 1994-2013



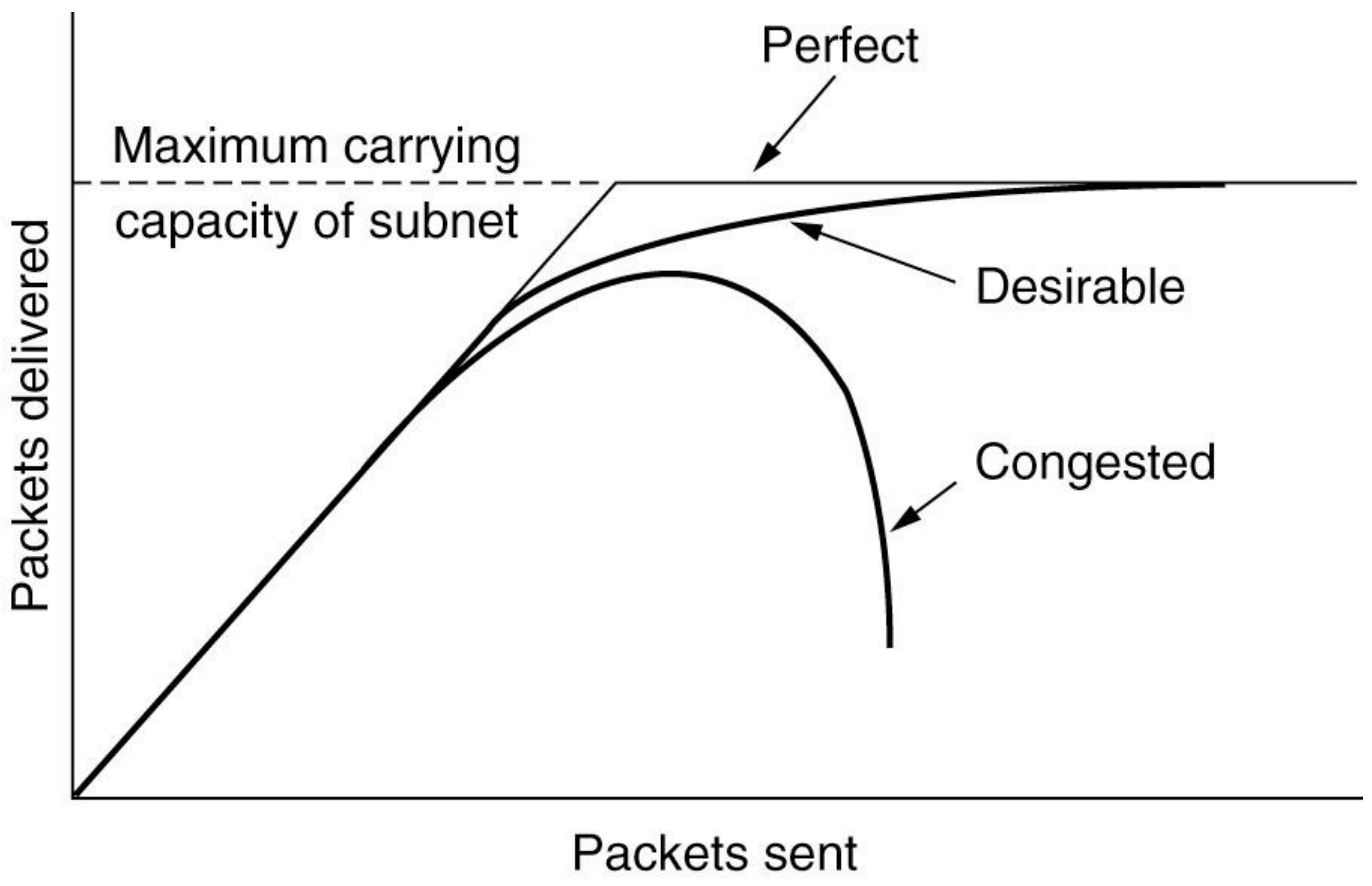
http://bgp.potaroo.net/as1221/bgp-active.html

Network Congestion

- (Sub-)Networks have limited bandwidth
- Injecting too many packets leads to
  - network congestion
  - network collapse



Congestion and capacity



Congestion Prevention

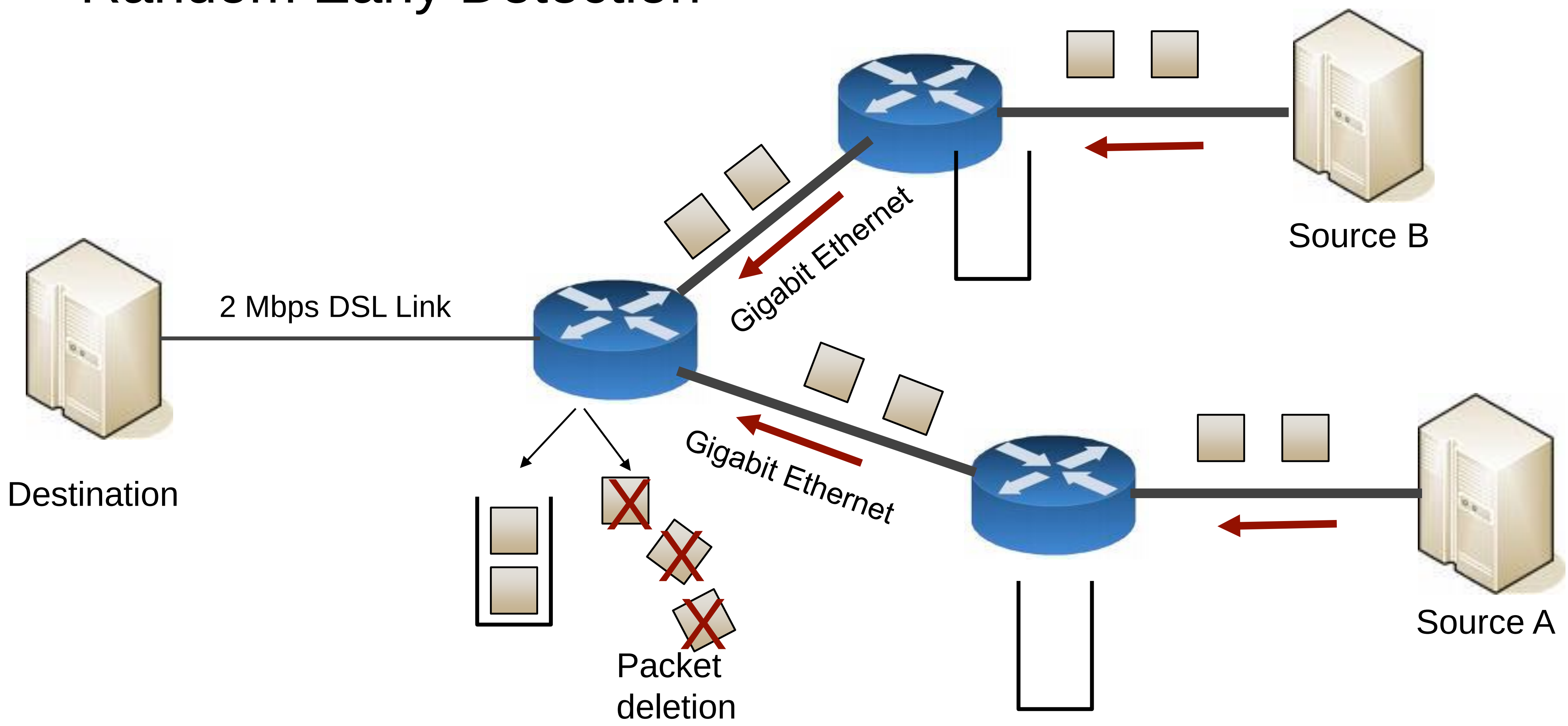
CoNe Freiburg

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

UNI FREIBURG

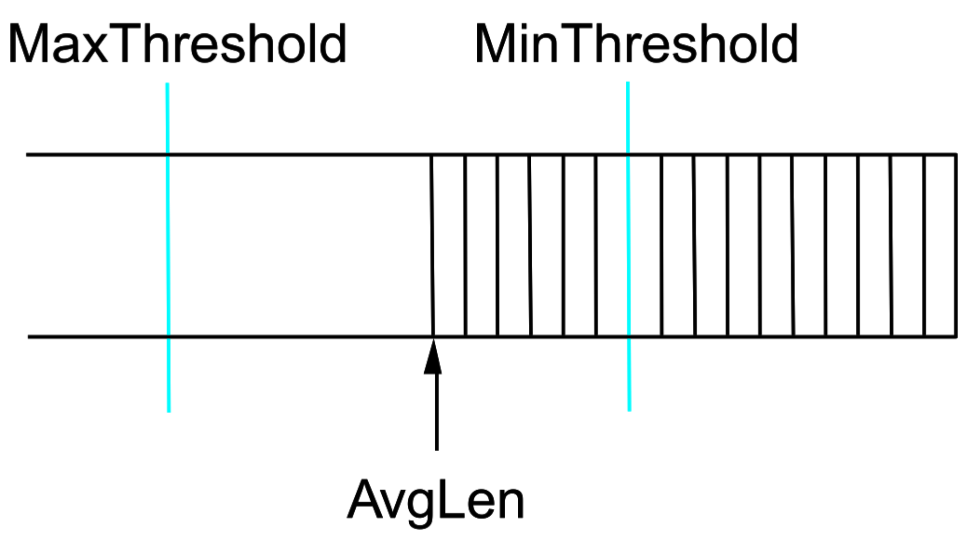
Congestion Prevention by Routers

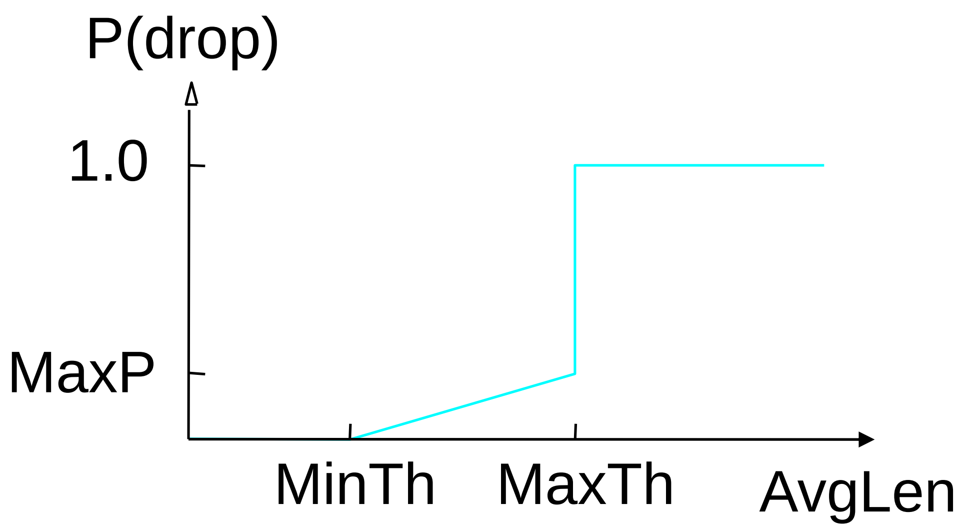
- IP Routers drop packets
  - Tail dropping
  - Random Early Detection



Random early detection (RED)

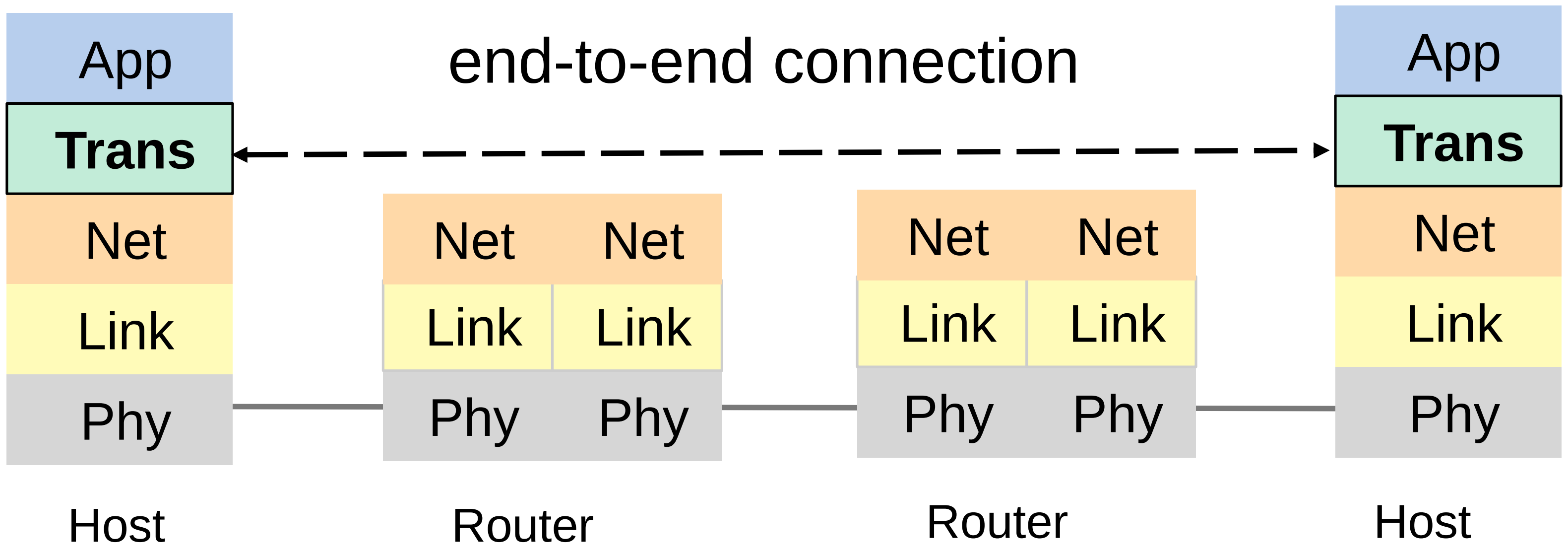
- Packet dropping probability grows with queue length
- Fairer than just “tail dropping”: the more a host transmits, the more likely it is that its packets are dropped





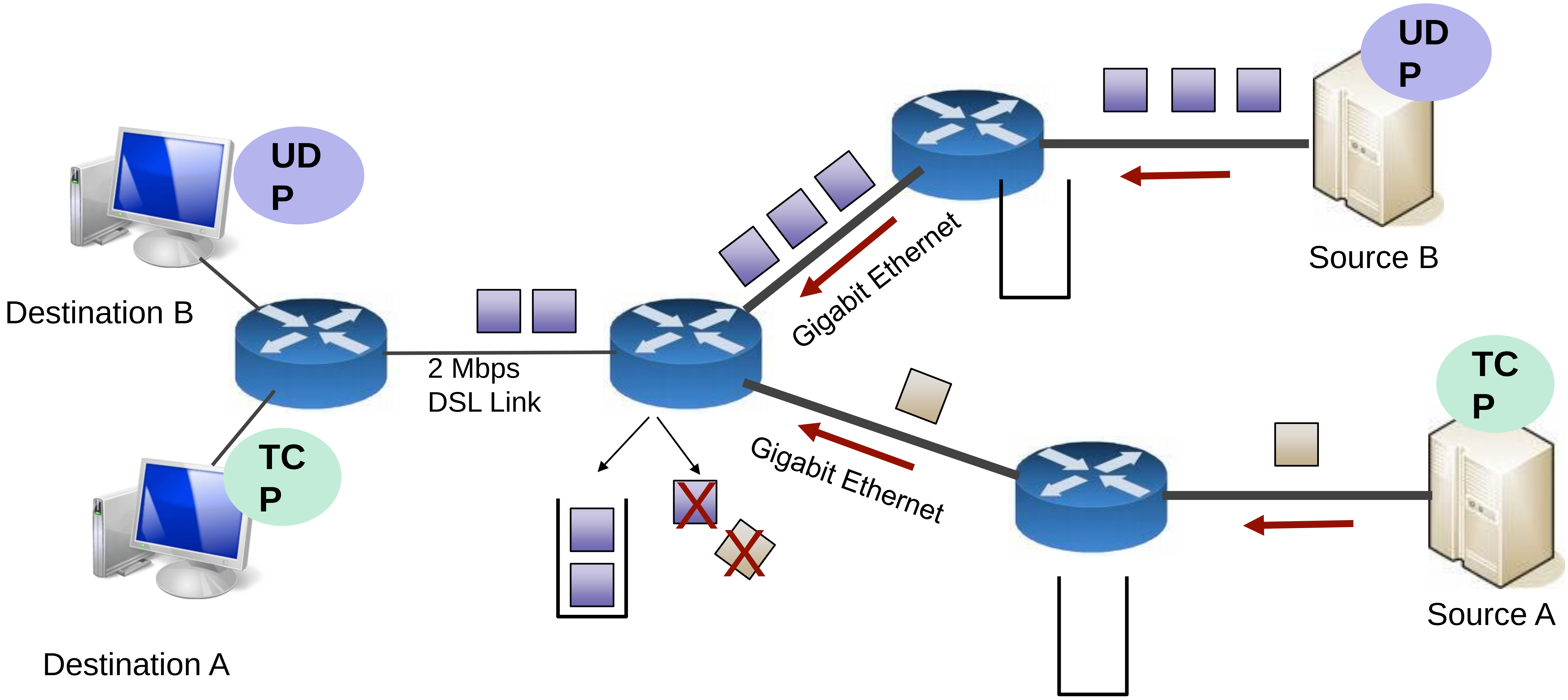
The Transport Layer

- TCP (Transmission Control Protocol
  - connection-oriented
  - delivers a stream of bytes
  - reliable and ordered
- UDP (User Datagram Protocol)
  - delivery of datagrams
  - connectionless, unreliable, unordered



TCP vs. UDP

- TCP reduces data rate
- UDP does not!



UDP-Header

- Port addresses
  - for parallel UDP connections
- Length
  - data + header length
- Checksum
  - for header and data

```
0      7 8     15 16    23 24    31
 +--------+--------+--------+--------+
 |     Source      |   Destination   |
 |      Port       |      Port       |
 +--------+--------+--------+--------+
 |                 |                 |
 |     Length      |    Checksum     |
 +--------+--------+--------+--------+
 |
 |          data octets ...
 +---------------- ...
```

The Transmission Control Protocol (TCP)

- Connection-oriented
- Reliable delivery of a byte stream
  - fragmentation and reassembly (*TCP segments*)
  - acknowledgements and retransmission
- In-order delivery, duplicate detection
  - sequence numbers
- Flow control and congestion control
  - window-based (receiver window, congestion window)

- challenge: IP (network layer) packets can be dropped, delayed, delivered out-of-order ...

CoNe Freiburg

TCP-Header

- Sequence number
  - number of the first byte in the segment
  - bytes are numbered modulo 2^{32}
- Acknowledge number
  - activated by ACK-Flag
  - number of the next data byte
    - = last sequence number + last amount of data
- Port addresses
  - for parallel TCP connections
- TCP Header length
  - data offset
- Check sum
  - for header and data

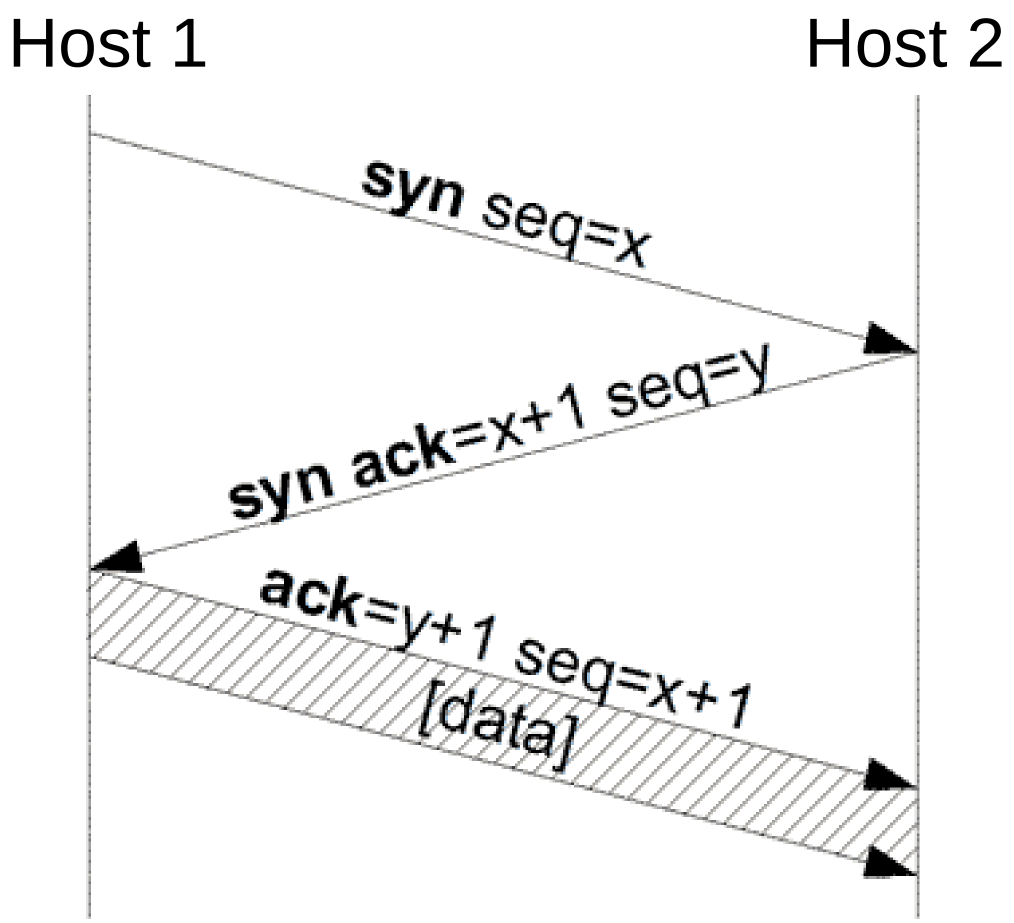
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

UNI FREIBURG

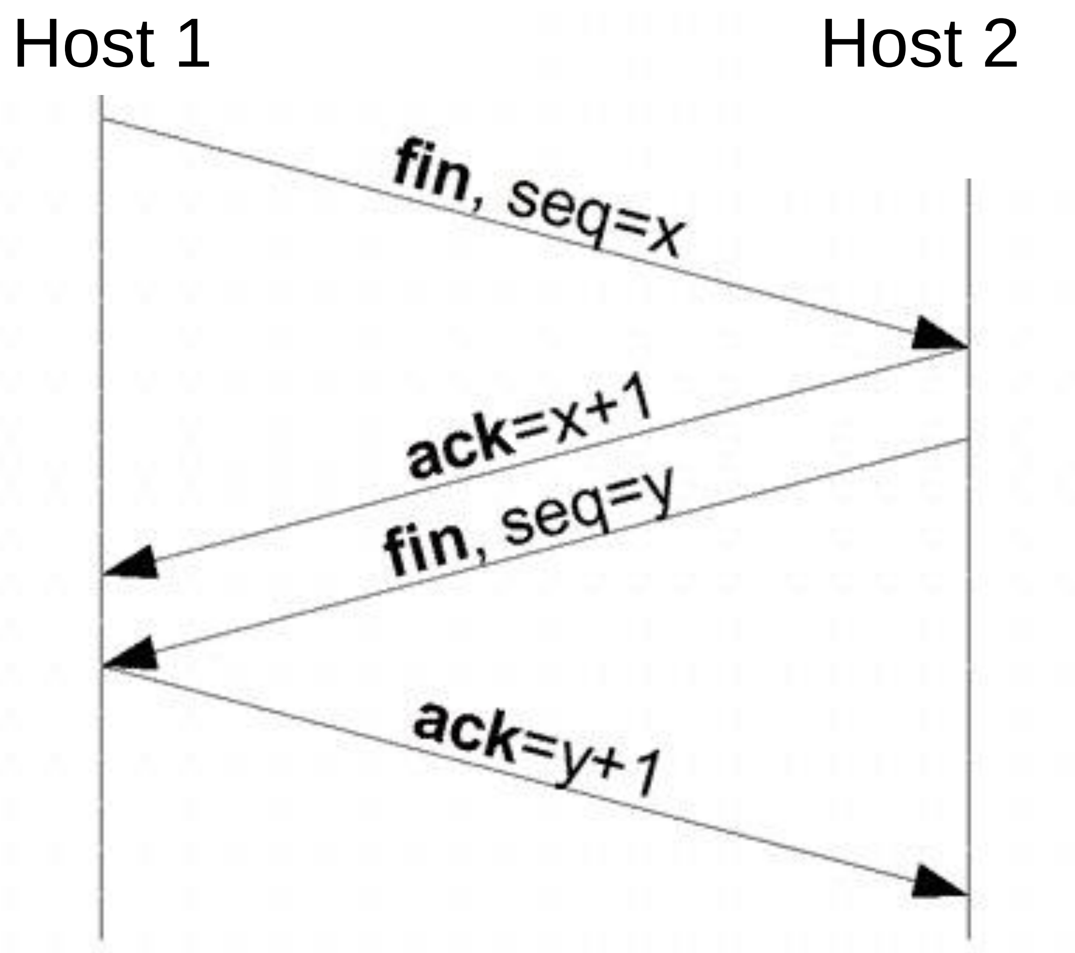
TCP Connections

- Connection establishment and teardown by 3-way handshake

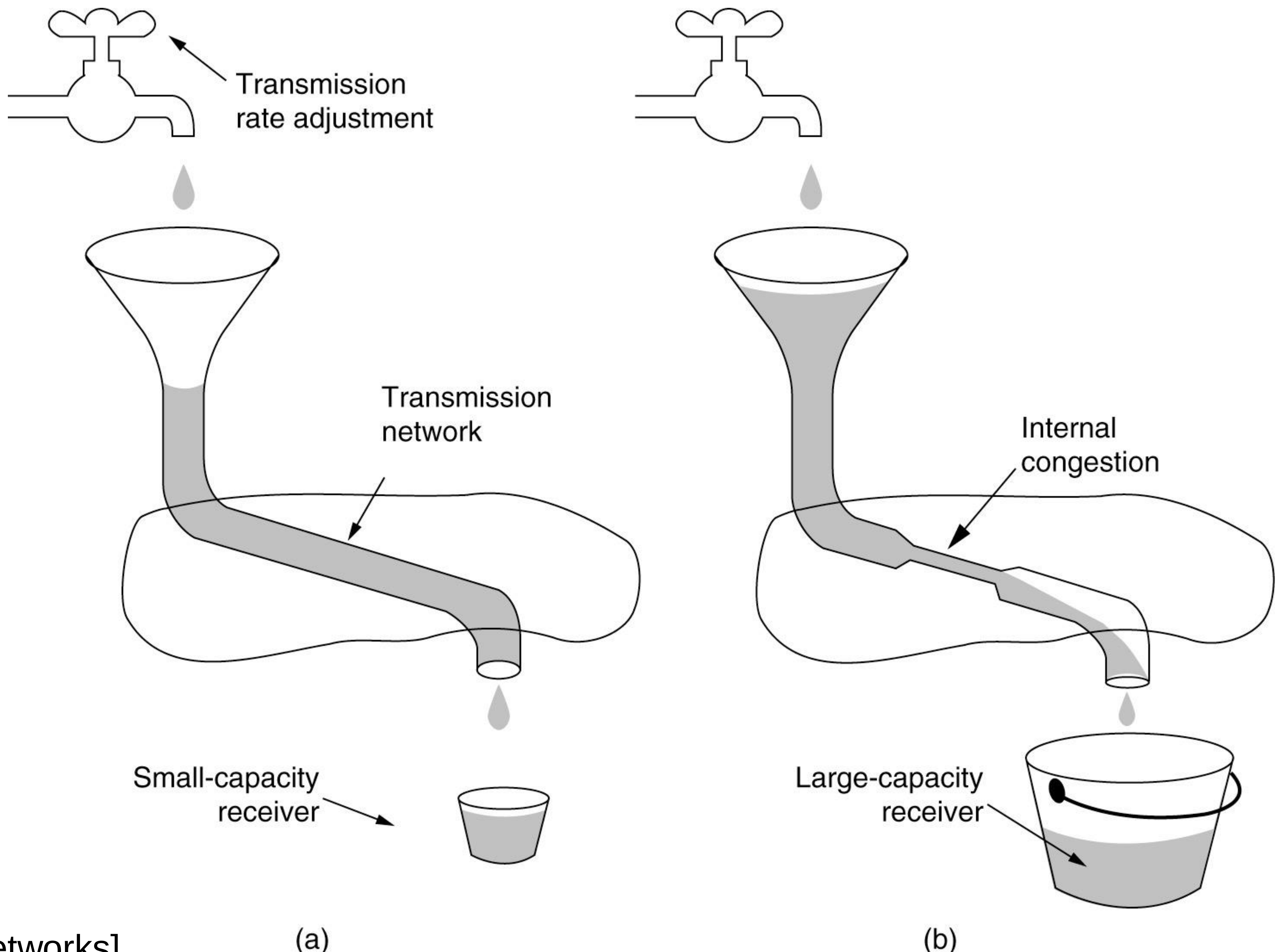
Connection establishment



Connection termination



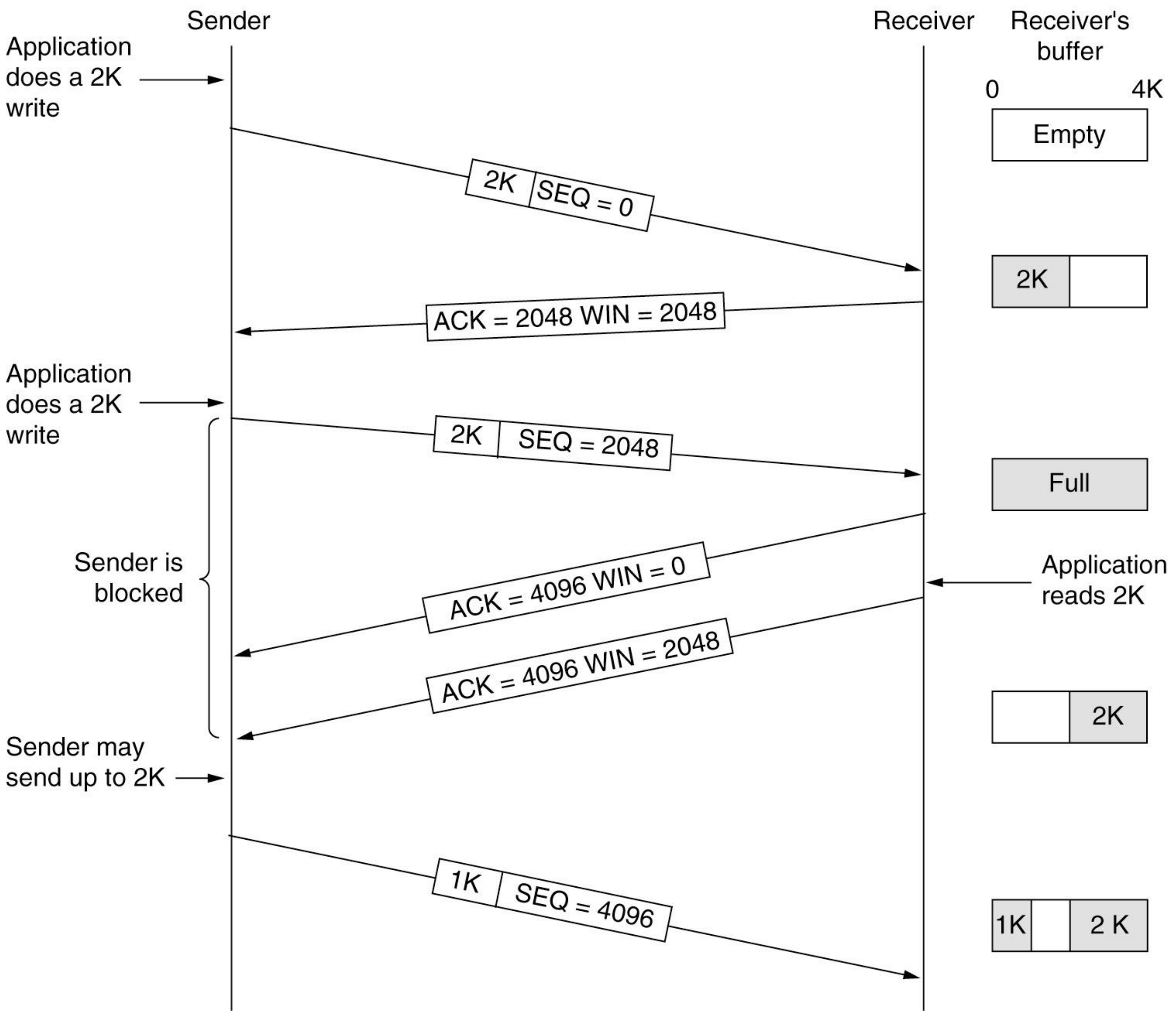
Flow control and congestion control



[Tanenbaum,
Computer Networks]

Flow Control

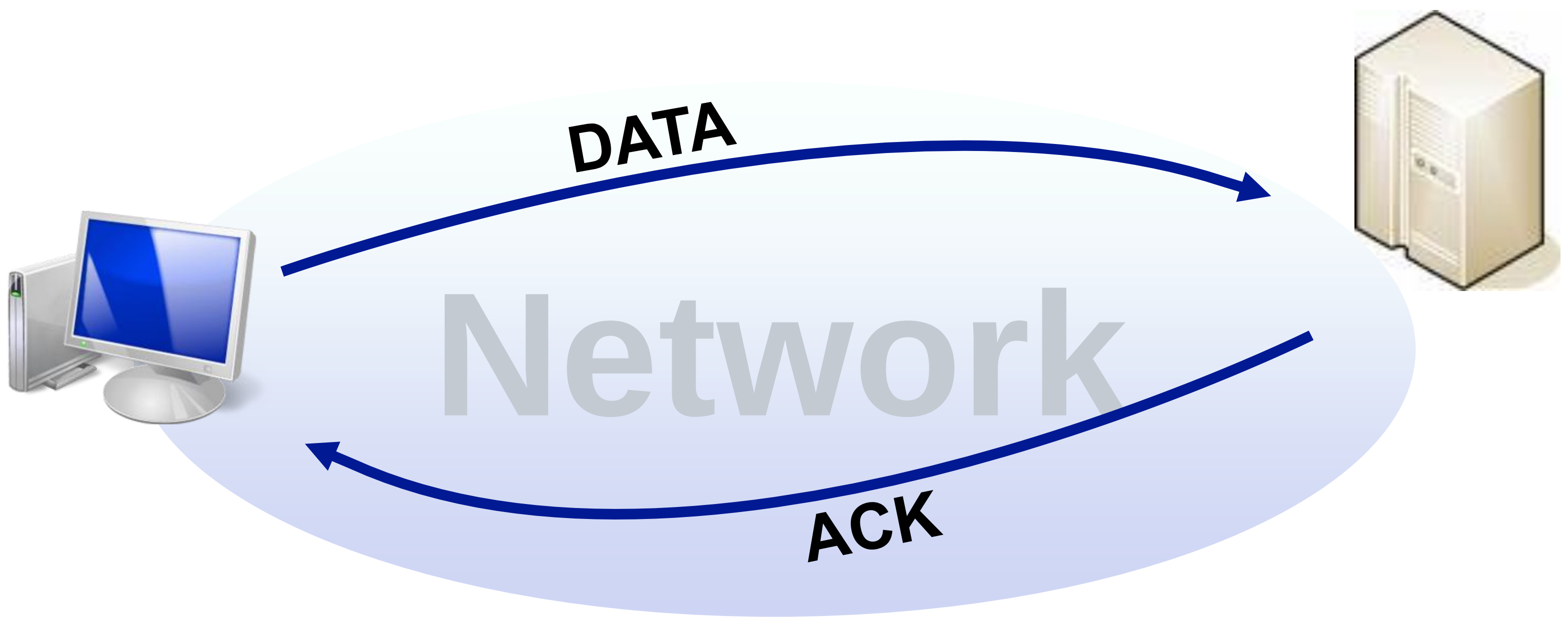
acknowledgements and window management



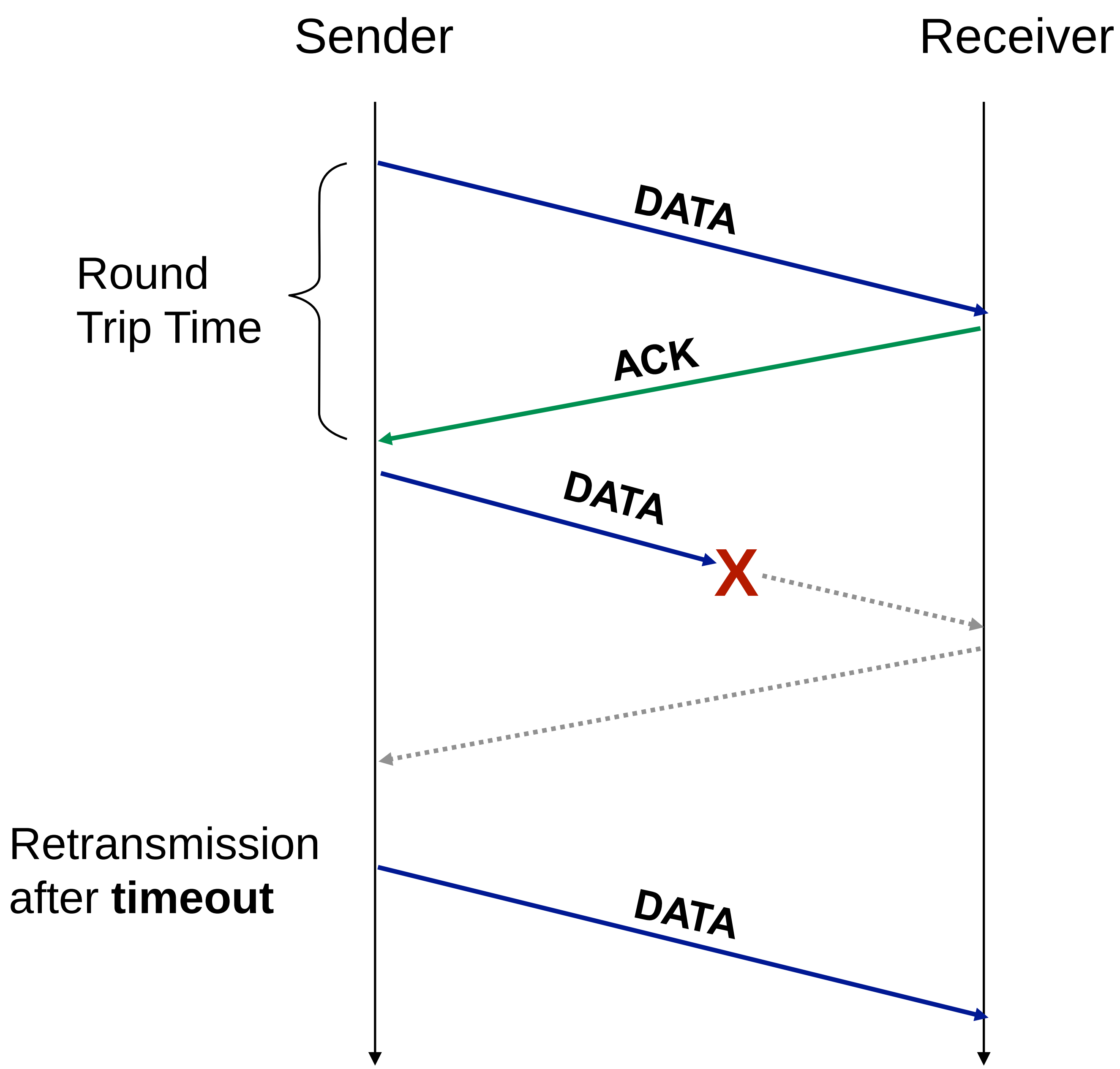
Retransmissions

- Retransmissions are triggered, if acknowledgements do not arrive
  ... but how to decide that?

- Measurement of the round trip time (RTT)



Retransmissions and RTT



Estimation of the Round Trip Time (RTT)

- If no acknowledgement arrives before expiry of the **Retransmission Timeout (RTO)**, the packet will be retransmitted
  - RTT not predictable, fluctuating

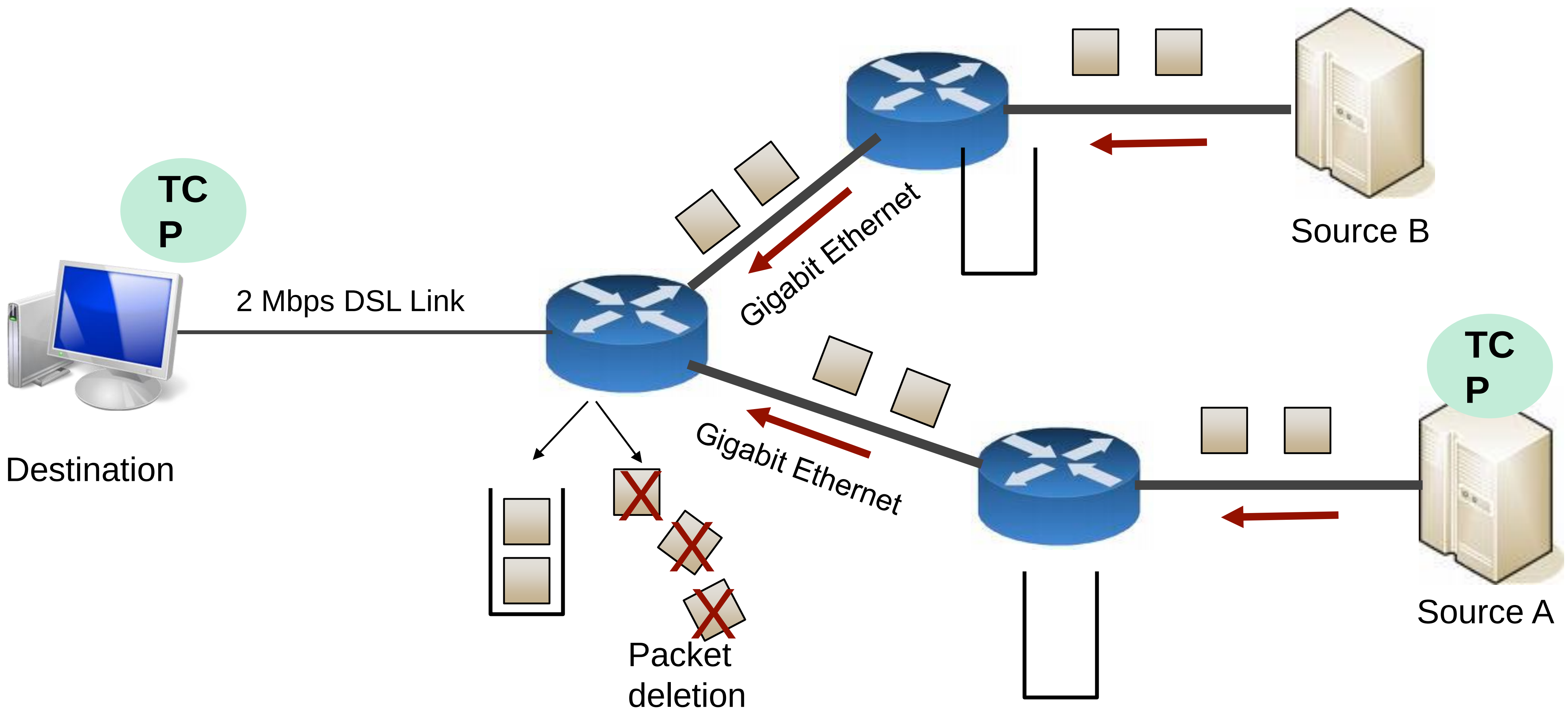
- **RTO derived from RTT estimation:**
  - RFC 793: (M := last RTT measurement)
    - RTT $\leftarrow \alpha$ RTT + (1-α) M, where α = 0,9
    - RTO $\leftarrow \beta$ RTT, where β = 2
  - Alternative by Jacobson 88 (using the deviation D):
    - D $\leftarrow \alpha'$ D + (1-α') |RTT - M|
    - RTT $\leftarrow \alpha$ RTT + (1-α) M
    - RTO $\leftarrow$ RTT + 4D

TCP - Algorithm of Nagle

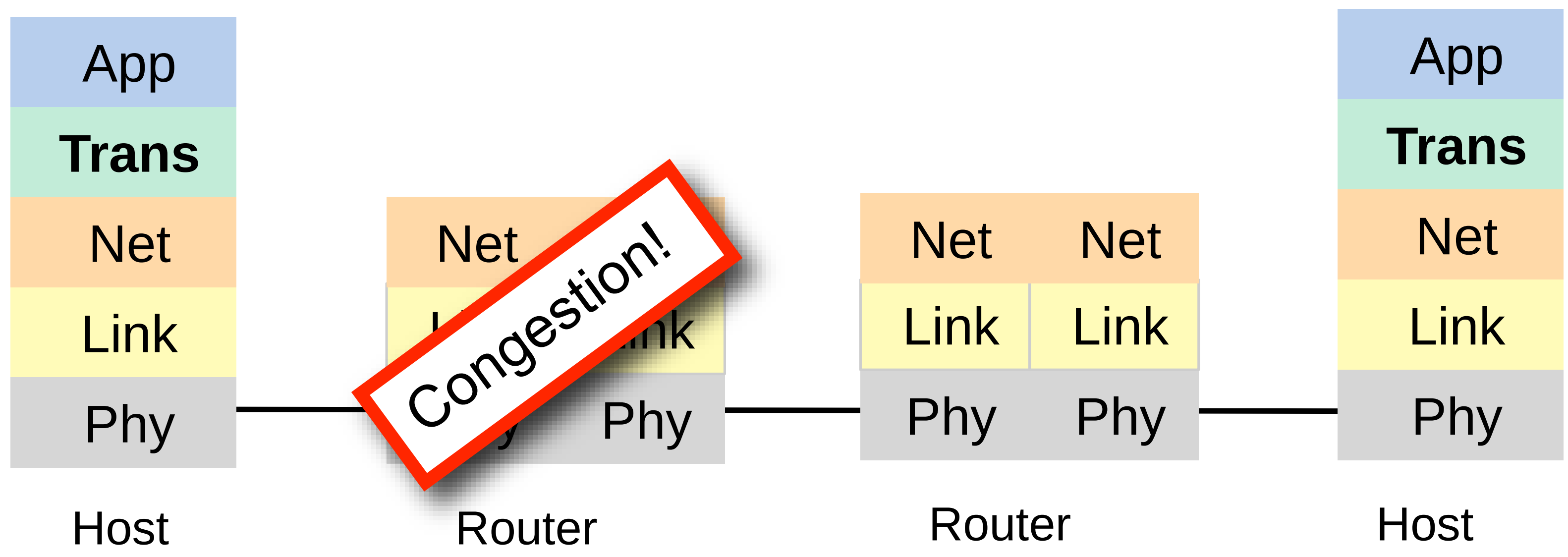
- How to ensure
  - small packages are shipped fast
  - yet, large packets are preferred
- Algorithm of Nagle
  - Small packets are not sent, as long as acks are still pending
    - Package is small, if data length <MSS
  - when the acknowledgment of the last packet arrives, the next one is sent
- Example:
  - terminal versus file transfer versus ftp
- Feature: self-clocking:
    - Quick link = many small packets
    - slow link = few large packets

Congestion revisited

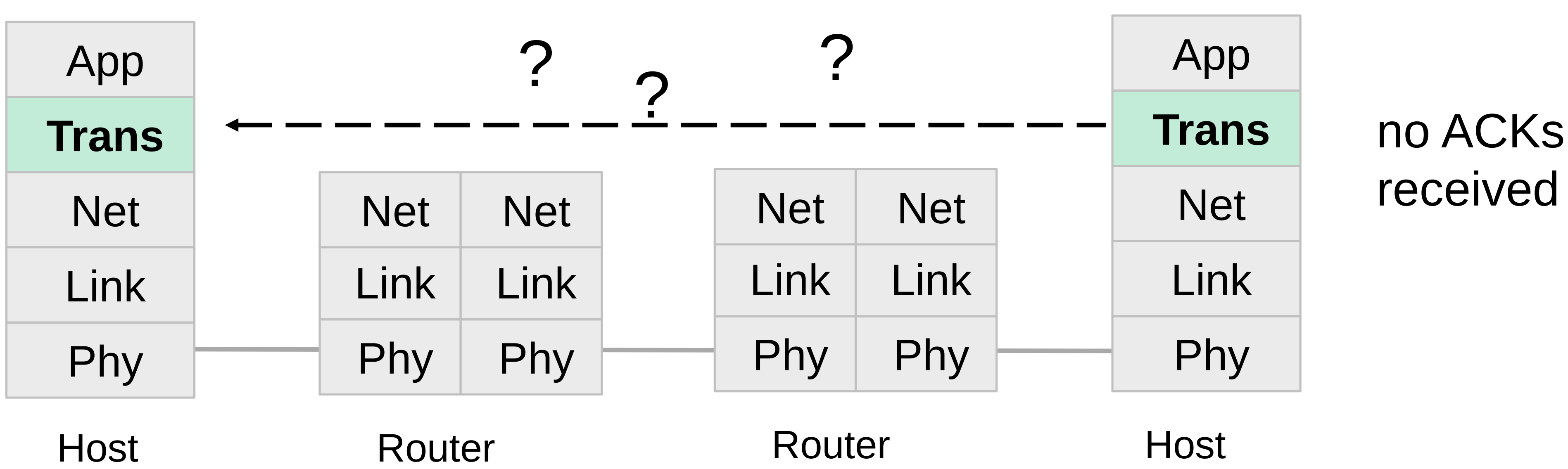
- IP Routers drop packets
- TCP has to react, e.g. lower the packet injection rate



Congestion revisited



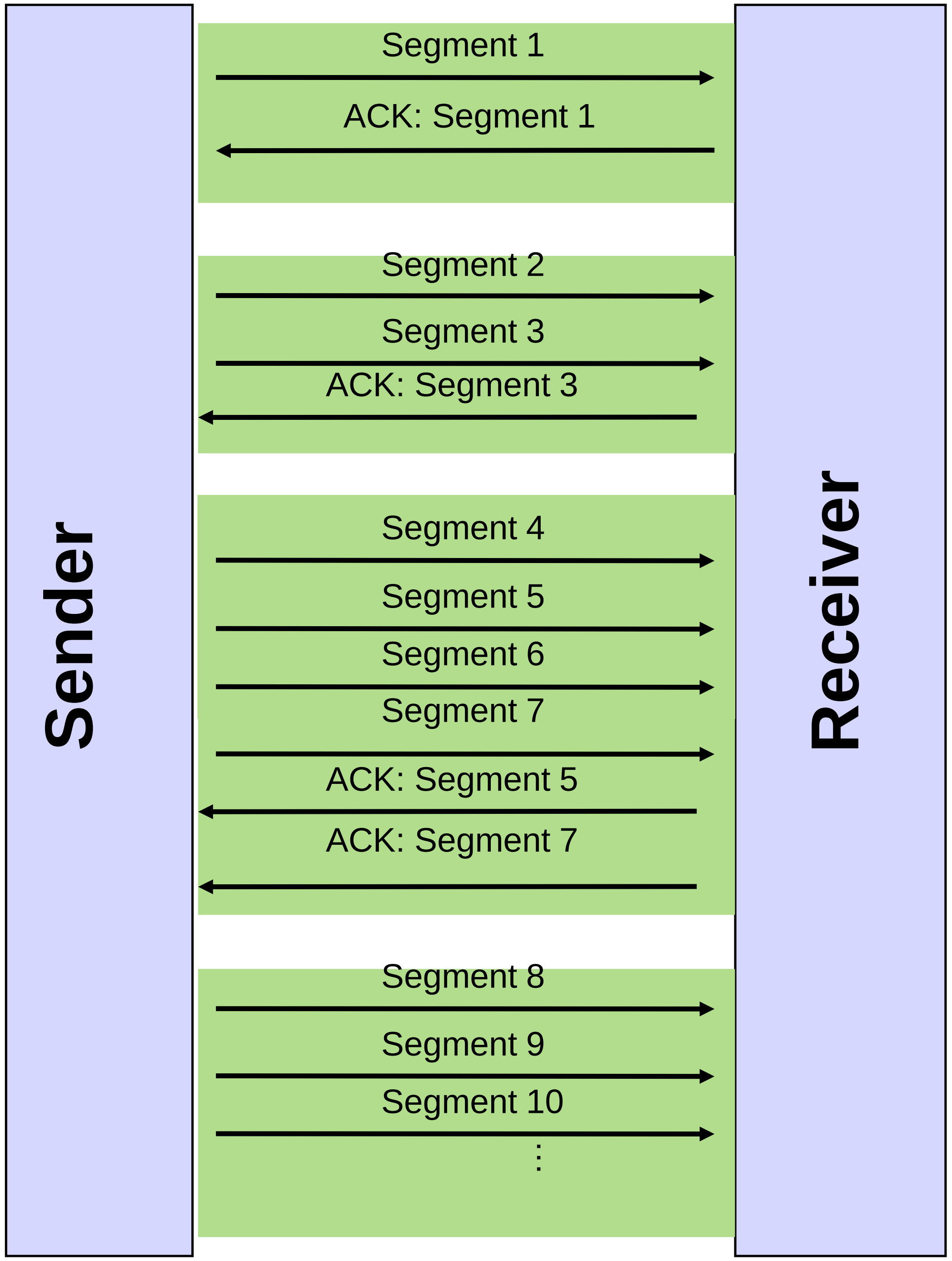
from a transport layer perspective:



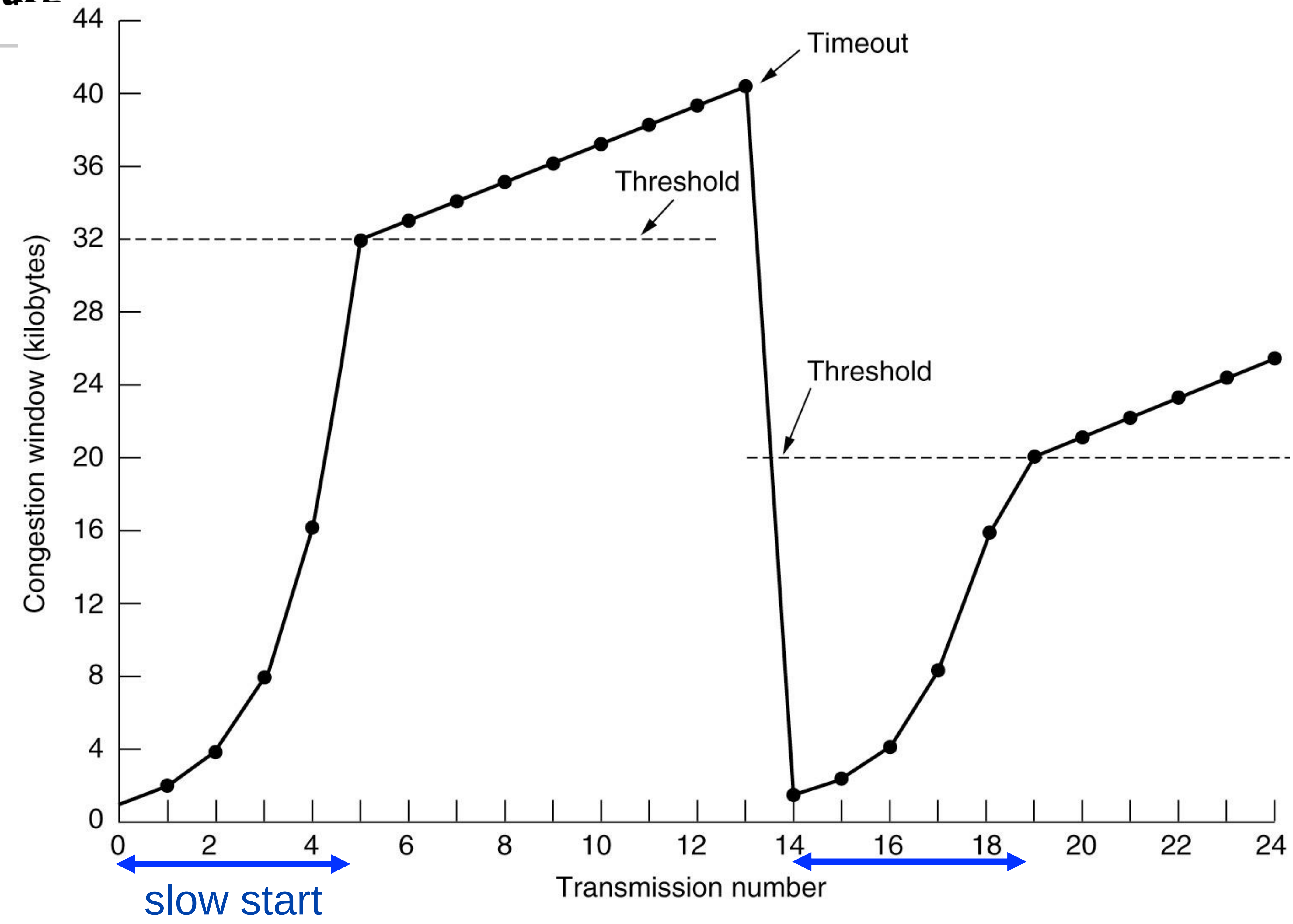
Data rate adaption and the congestion window

- Sender does not use the maximum segment size in the beginning

- Congestion window (cwnd)
  - used on the sender size
  - sending window: min {wnd,cwnd} (wnd = receiver window)
  - S: segment size
  - Initialization:
    - cwnd $\leftarrow$ S
  - For each received acknowledgement:
    - cwnd $\leftarrow$ cwnd + S
  - ...until a packet remains unacknowledged



Slow Start of TCP Tahoe



TCP Tahoe’s slow start

CoNe Freiburg

- **TCP Tahoe, Jacobson 88:**
  - Congestion window (cwnd)
  - Slow Start Threshold (ssthresh)
  - S = maximum segment size
- **Initialization (after connection establishment):**
  - cwnd $\leftarrow$ S   ssthresh $\leftarrow$ 65535
- **If a packet is lost (no acknowledgement within RTO):**
  - multiplicative decrease of ssthresh
    cwnd $\leftarrow$ S   ssthresh $\leftarrow \max\left\{2 \times S, \frac{\min\{cwnd, wnd\}}{2}\right\}$
- **If a segment is acknowledged and cwnd ≤ ssthresh then**
  - slow start: cwnd $\leftarrow$ cwnd + S
- **If a segment is acknowledged and cwnd > ssthresh, then**

  cwnd $\leftarrow$ cwnd + S/cwnd

x: # Packets per RTT

x $\leftarrow$ 1 | y $\leftarrow$ max

x $\leftarrow$ 1 | y $\leftarrow$ x/2

x $\leftarrow$ 2⊕x, until x = y

x $\leftarrow$ x +1

UNI FREIBURG

Fast Retransmit and Fast Recovery

- TCP Tahoe [Jacobson 1988]:
  - If only one packet is lost
    - retransmit and use the rest of the window
    - Slow Start
  - Fast Retransmit
    - after three duplicate ACKs, retransmit Packet, start with Slow Start

- TCP Reno [Stevens 1994]
  - After Fast Retransmit:
    - ssthresh $\leftarrow$ min(wnd,cwnd)/2
    - cwnd $\leftarrow$ ssthresh + 3 S
  - Fast recovery after Fast retransmit
    - Increase window size by each single acknowledgement
    - cwnd $\leftarrow$ cwnd + S
  - Congestion avoidance: if P+x is acknowledged:
    - cwnd $\leftarrow$ ssthresh

$y \leftarrow x/2$
$x \leftarrow y + 3$

The AIMD principle

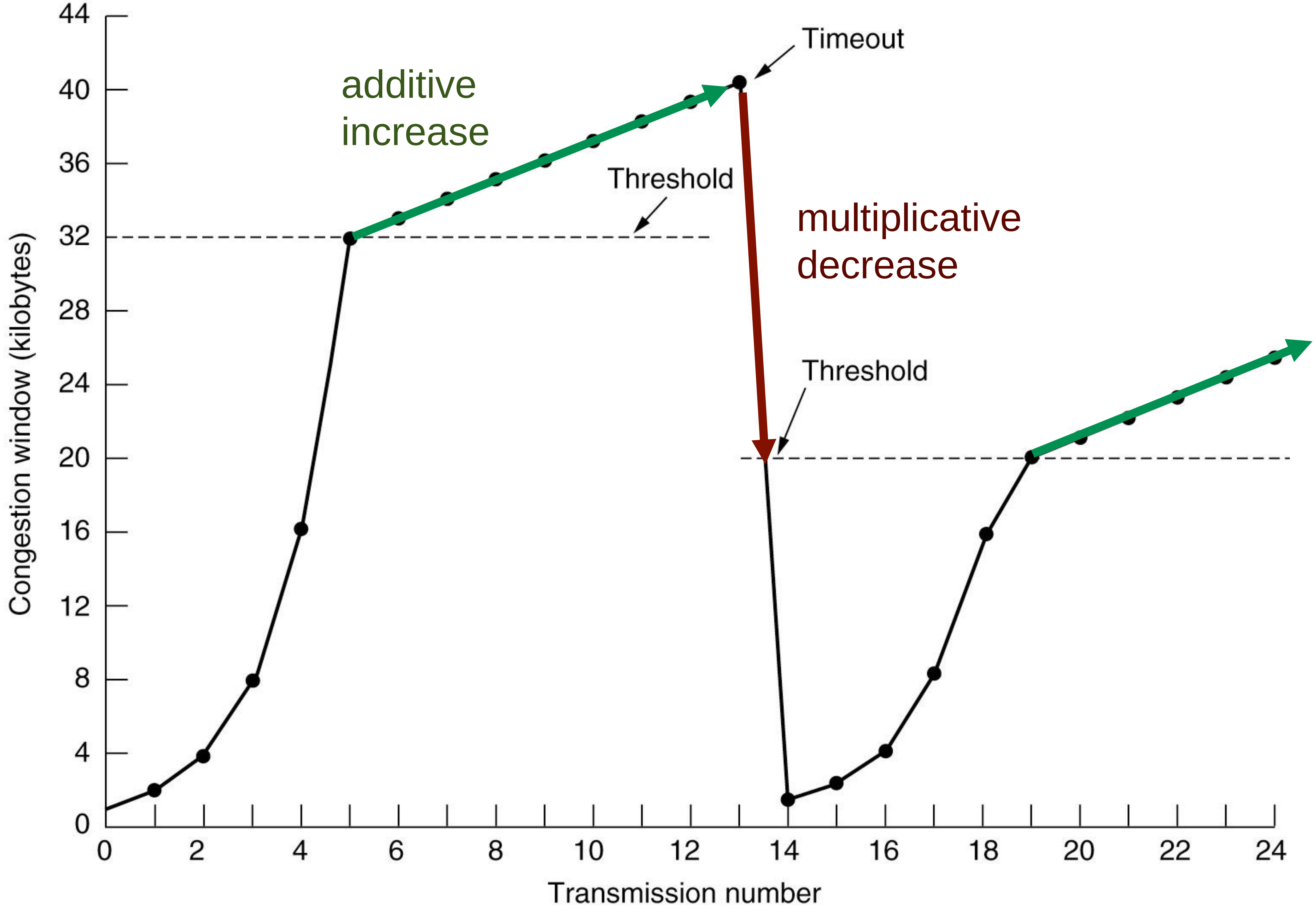
- TCP uses basically the following mechanism to adapt the data rate x (#packets sent per RTT):

  - Initialization: $x \leftarrow 1$

  - on packet loss: multi $x \leftarrow x/2$ crease (MD)

  - if the acknowledgement for a segment arrives, perform additive increase (AI) $x \leftarrow x+1$

AIMD

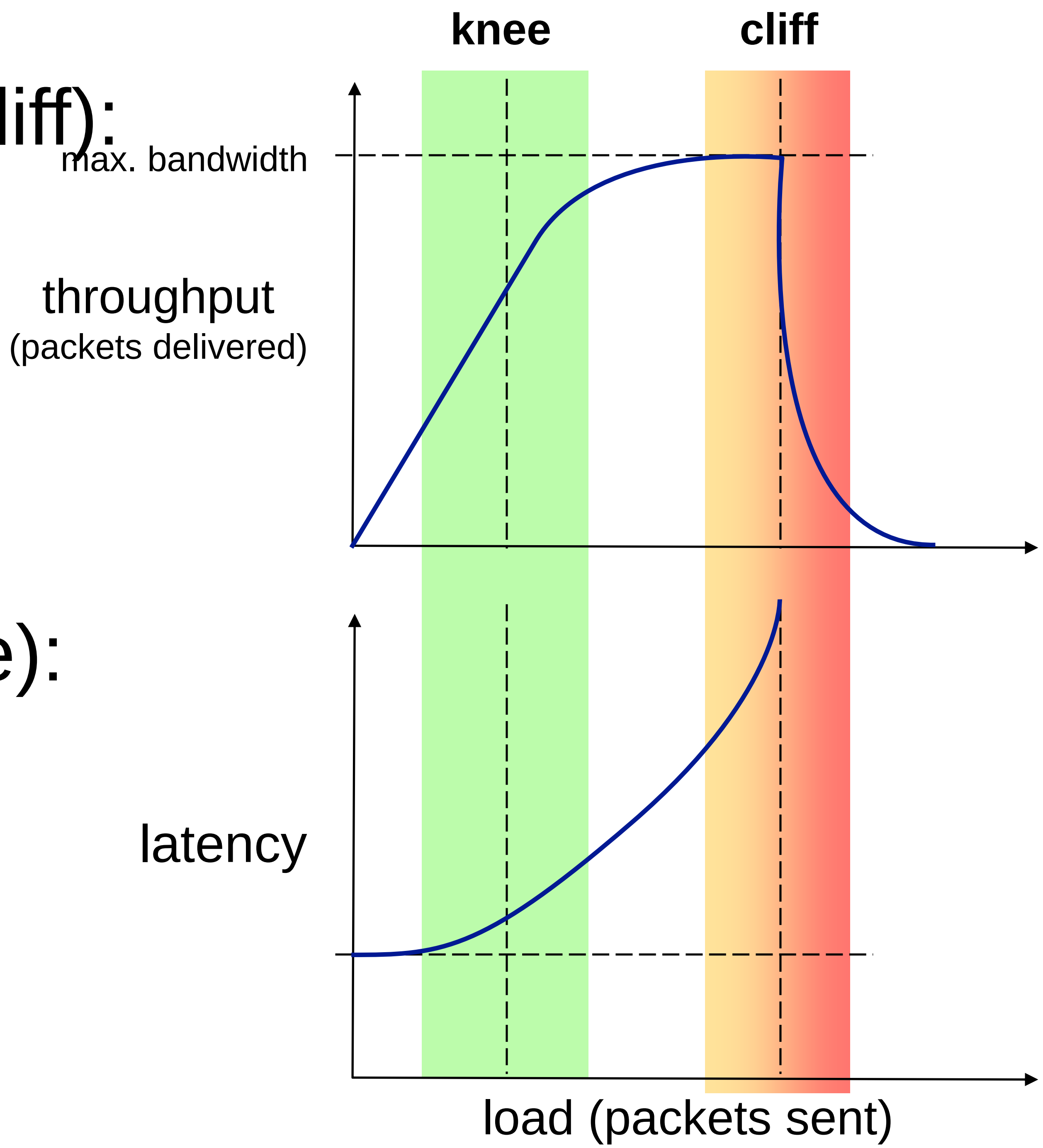


additive
increase
multiplicative
decrease
Timeout
Threshold
Threshold
Congestion window (kilobytes)
Transmission number
0
4
8
12
16
20
24
28
32
36
40
44
0
2
4
6
8
10
12
14
16
18
20
22
24

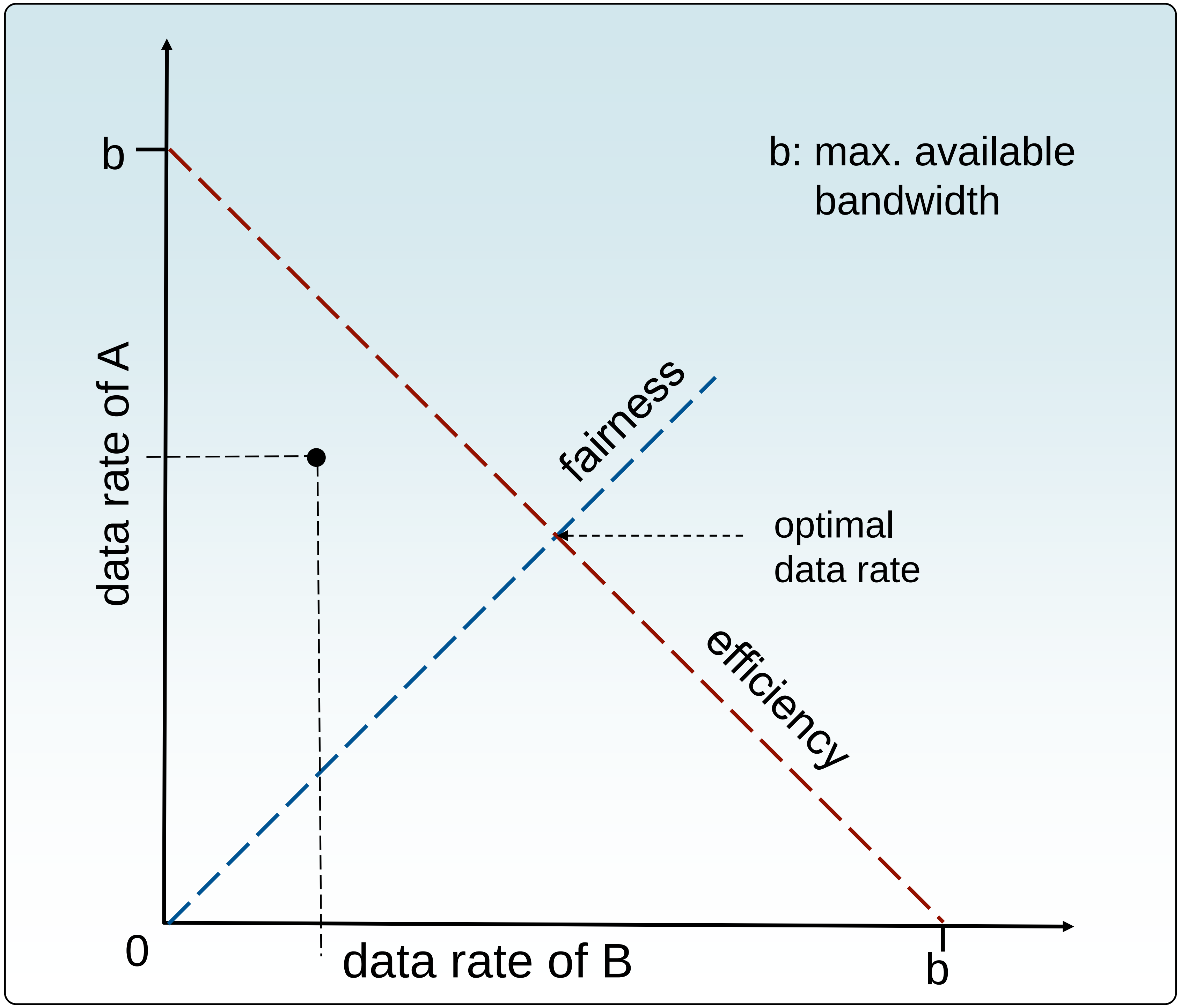
Throughput and Latency

- Congested situation (cliff):
  - high load
  - low throughput
  - all data packets are lost

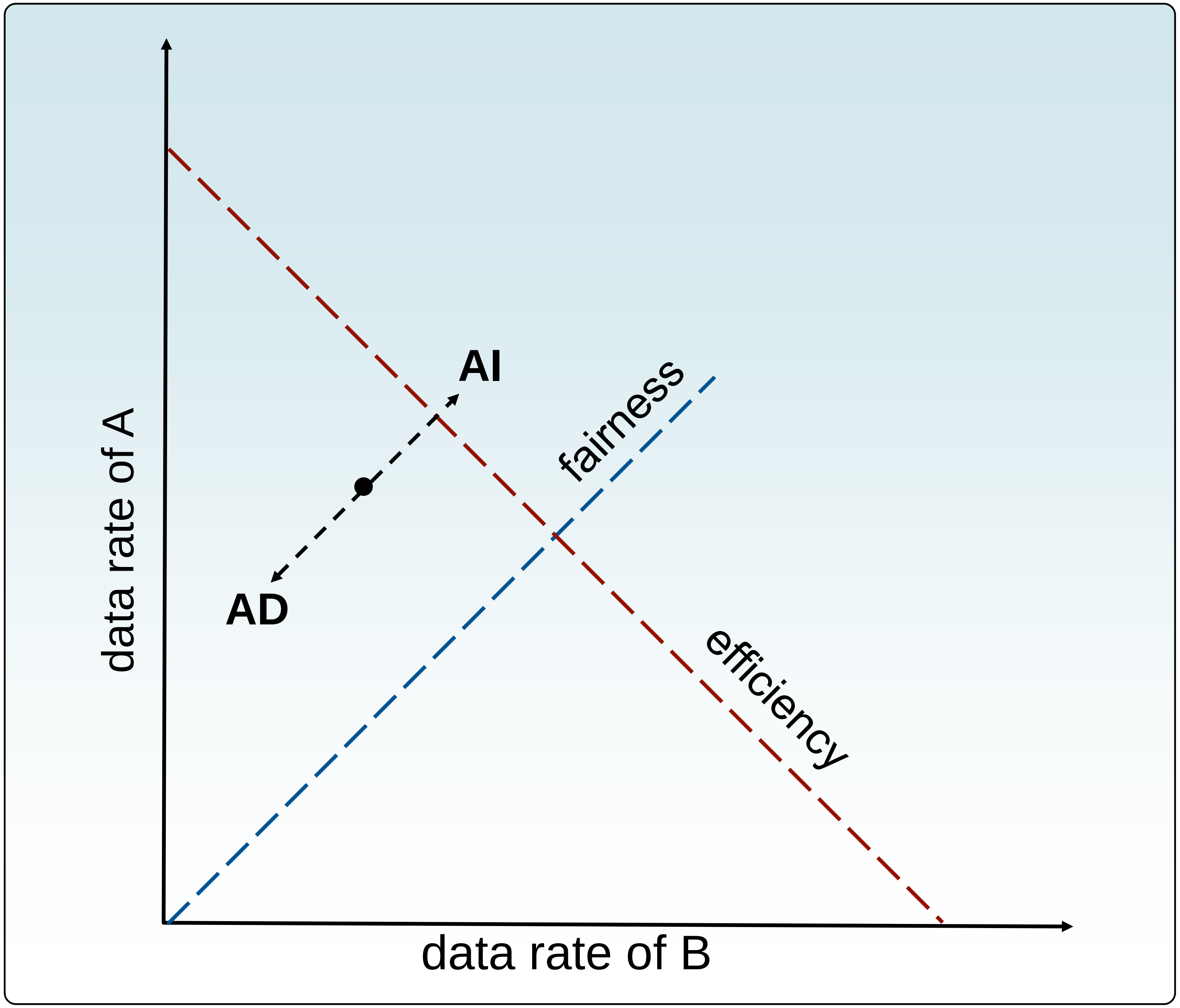
- Desired situation (knee):
  - high load
  - high throughput
  - few data packets get lost



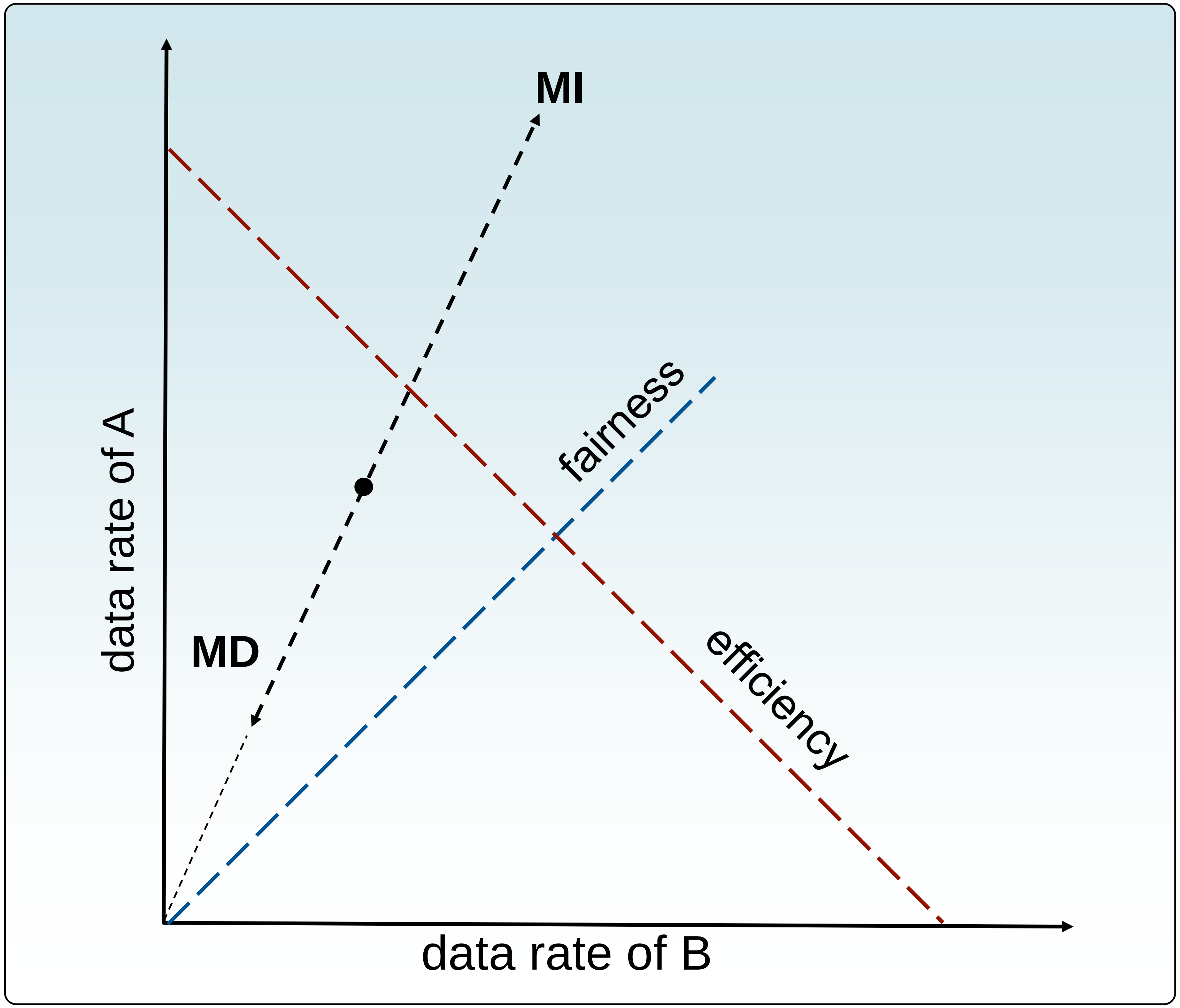
Vector diagram for 2 participants



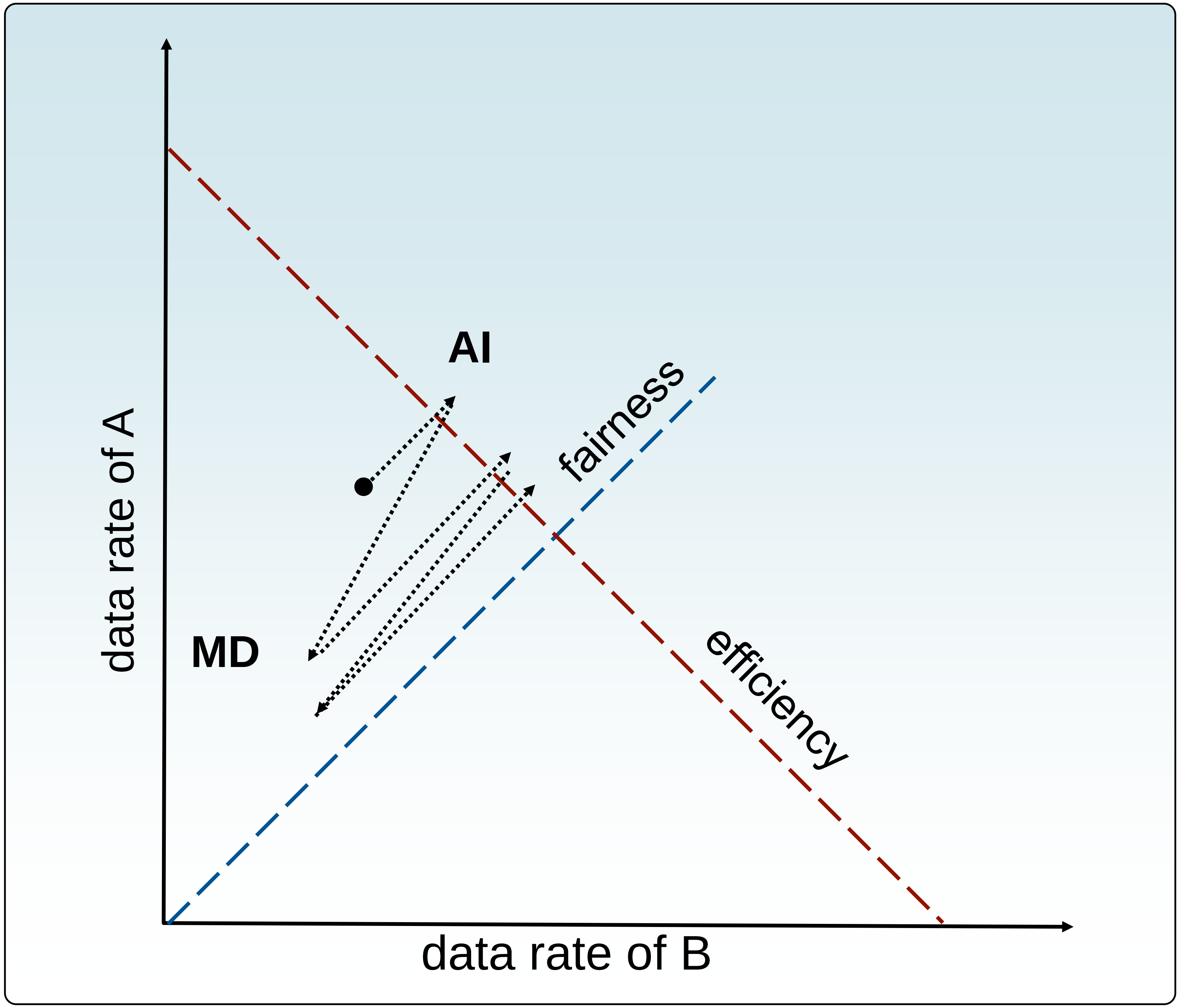
AIAD Additive Increase/ Additive Decrease



MIMD: Multiplicative Incr./ Multiplicative Decrease



AIMD: Additively Increase/ Multiplicatively Decrease



TCP - Conclusion

- Connection-oriented, reliable,
  in-order delivery of a byte stream
- Flow control and congestion control
  - Fairness among TCP streams
  - Unfair behavior of other protocols, e.g. UDP
  - Impact on latency
  - Tweaking the congestion avoidance mechanism has an impact on other applications

Peer-to-Peer Networks

13 Internet – The Underlay Network

Christian Ortolf
Technical Faculty
Computer-Networks and Telematics
University of Freiburg